

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Applicants: Takashi YOKOKAWA et al.  
International Application No.: PCT/JP04/005551  
International Filing Date: April 19, 2004  
For: DECODING METHOD, DECODING APPARATUS,  
AND PROGRAM

745 Fifth Avenue  
New York, NY 10151

**EXPRESS MAIL**

Mailing Label Number: EV206809936US

Date of Deposit: January 12, 2005

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" Service under 37 CFR 1.10 on the date indicated above and is addressed to Mail Stop PCT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Barnet Shindman  
(Typed or printed name of person mailing paper or fee)

Barnet Shindman  
(Signature of person mailing paper or fee)

**CLAIM OF PRIORITY UNDER 37 C.F.R. § 1.78(a)(2)**

Mail Stop PCT  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

Pursuant to 35 U.S.C. 119, this application is entitled to a claim of priority to Japan  
Application No. 2003-133942 filed 13 May 2003.

Respectfully submitted,

FROMMER LAWRENCE & HAUG LLP  
Attorneys for Applicants

By: William S. Frommer

William S. Frommer  
Reg. No. 25,506  
Tel. (212) 588-0800

日本国特許庁  
JAPAN PATENT OFFICE

19. 4. 2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 5月13日  
Date of Application:

出願番号 特願2003-133942  
Application Number:  
[ST. 10/C]: [JP2003-133942]

出願人 ソニー株式会社  
Applicant(s):

REC'D 13 MAY 2004

WIPO

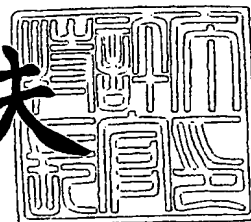
PCT

PRIORITY DOCUMENT  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

2004年 2月19日

特許庁長官  
Commissioner,  
Japan Patent Office

今井康夫



BEST AVAILABLE COPY

【書類名】 特許願

【整理番号】 0390404503

【提出日】 平成15年 5月13日

【あて先】 特許庁長官殿

【国際特許分類】 H03M 13/03

【発明者】

    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
    内

    【氏名】 横川 峰志

【発明者】

    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
    内

    【氏名】 菅 真紀子

【発明者】

    【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社  
    内

    【氏名】 飯田 康博

【発明者】

    【住所又は居所】 神奈川県横浜市保土ヶ谷区新井町 2 2 0 の 2 0

    【氏名】 菊池 敦

【特許出願人】

    【識別番号】 000002185

    【氏名又は名称】 ソニー株式会社

【代理人】

    【識別番号】 100082131

    【弁理士】

    【氏名又は名称】 稲本 義雄

    【電話番号】 03-3369-6479

【手数料の表示】

【予納台帳番号】 032089

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9708842

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 復号方法および復号装置、並びにプログラム

【特許請求の範囲】

【請求項 1】 LDPC(Low Density Parity Check)符号の復号方法であって、元の検査行列に対して、行置換と列置換のうち的一方または両方を行って得られる変換検査行列を用いて、前記LDPC符号を復号する復号ステップを備えることを特徴とする復号方法。

【請求項 2】 請求項 1 に記載の復号方法であって、 $P \times P$ の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上が 0 になった行列である準単位行列、前記単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、前記単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または  $P \times P$ の 0 行列を構成行列として、前記変換検査行列は、複数の前記構成行列の組合せで表されることを特徴とする復号方法。

【請求項 3】 請求項 1 に記載の復号方法であって、受信した前記LDPC符号の符号系列に対して、前記元の検査行列に対して行われた列置換と同一の列置換を行い、置換符号系列を出力する符号系列置換ステップをさらに備え、

前記復号ステップにおいて、前記変換検査行列と、前記置換符号系列とを用いて、前記符号系列を復号することを特徴とする復号方法。

【請求項 4】 LDPC(Low Density Parity Check)符号の復号装置であって、元の検査行列に対して、行置換と列置換のうち的一方または両方を行って得られる変換検査行列を用いて、前記LDPC符号を復号する復号手段を備えることを特徴とする復号装置。

【請求項 5】 請求項 4 に記載の復号装置であって、 $P \times P$ の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上が 0 になった行列である準単位行列、前記単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、前記単位行列、準単位行列、もしくはシ

フト行列のうちの複数の和である和行列、または $P \times P$ の0行列を構成行列として、前記変換検査行列は、複数の前記構成行列の組合せで表されることを特徴とする復号装置。

【請求項6】 請求項5に記載の復号装置であって、前記復号手段は、

前記LDPC符号の復号のためのP個のチェックノードの演算を同時に行うチェックノード計算手段と、

前記LDPC符号の復号のためのP個のバリエブルノードの演算を同時に行うバリエブルノード計算手段と

を有する

ことを特徴とする復号装置。

【請求項7】 請求項6に記載の復号装置であって、

前記チェックノード計算手段は、チェックノードの演算を行うP個のチェックノード計算器を有し、

前記バリエブルノード計算手段は、バリエブルノードの演算を行うP個のバリエブルノード計算器を有する

ことを特徴とする復号装置。

【請求項8】 請求項6に記載の復号装置であって、

前記復号手段は、前記P個のチェックノードの演算、または前記P個のバリエブルノードの演算の結果得られるP個の枝に対応するメッセージデータを同時に読み書きするメッセージ記憶手段をさらに有する

ことを特徴とする復号装置。

【請求項9】 請求項8に記載の復号装置であって、

前記メッセージ記憶手段は、チェックノード演算時に読み出される枝に対応するメッセージデータを、前記変換検査行列の1を行方向に詰めるように格納することを特徴とする復号装置。

【請求項10】 請求項8に記載の復号装置であって、

前記メッセージ記憶手段は、バリエブルノード演算時に読み出される枝に対応するメッセージデータを、前記変換検査行列の1を列方向に詰めるように格納す

る

ことを特徴とする復号装置。

【請求項 11】 請求項 8 に記載の復号装置であって、

前記メッセージ記憶手段は、前記変換検査行列を表す構成行列のうちの、重みが 2 以上の構成行列について、その構成行列を、重みが 1 の単位行列、準単位行列、またはシフト行列の和の形で表現したときの、その重みが 1 の単位行列、準単位行列、またはシフト行列に属する P 個の枝に対応するメッセージを、同一のアドレスに格納する

ことを特徴とする復号装置。

【請求項 12】 請求項 8 に記載の復号装置であって、

前記メッセージ記憶手段は、行数/P 個の FIFO と、列数/P 個の FIFO とで構成され、

前記行数/P 個の FIFO と列数/P 個の FIFO は、それぞれ、前記検査行列の行と列の重みに対応するワード数を有する

ことを特徴とする復号装置。

【請求項 13】 請求項 8 に記載の復号装置であって、

前記メッセージ記憶手段は、RAM(Random Access Memory)で構成され、

前記 RAM は、前記メッセージデータを、読み出される順番に詰めて格納し、格納位置順に読み出す

ことを特徴とする復号装置。

【請求項 14】 請求項 6 に記載の復号装置であって、

前記復号手段は、受信情報を格納するとともに、P 個の前記受信情報を同時に読み出す受信情報記憶手段をさらに有する

ことを特徴とする復号装置。

【請求項 15】 請求項 14 に記載の復号装置であって、

前記受信情報記憶手段は、前記受信情報を、前記バリエブルノードの演算に必要となる順番に読み出すことができるように格納する

ことを特徴とする復号装置。

【請求項 16】 請求項 6 に記載の復号装置であって、

前記復号手段は、前記P個のチェックノードの演算、または前記P個のバリエーションノードの演算の結果得られるメッセージをサイクリックシフトするサイクリックシフト手段をさらに有する

ことを特徴とする復号装置。

【請求項 17】 請求項 16 に記載の復号装置であって、  
前記サイクリックシフト手段は、バレルシフタで構成される  
ことを特徴とする復号装置。

【請求項 18】 請求項 4 に記載の復号装置であって、  
受信した前記LDPC符号の符号系列に対して、前記元の検査行列に対して行われた列置換と同一の列置換を行い、置換符号系列を出力する符号系列置換手段をさらに備え、

前記復号手段は、前記変換検査行列と、前記置換符号系列とを用いて、前記符号系列を復号する

ことを特徴とする復号装置。

【請求項 19】 請求項 18 に記載の復号装置であって、  
前記復号手段の出力に対して、前記元の検査行列に対して行われた列置換の逆置換を行い、最終的な復号結果を出力する逆置換手段をさらに備える

ことを特徴とする復号装置。

【請求項 20】 LDPC (Low Density Parity Check) 符号の復号をコンピュータに行わせるプログラムであって、

元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、前記LDPC符号を復号する復号ステップを備える

ことを特徴とするプログラム。

#### 【発明の詳細な説明】

#### 【0001】

#### 【発明の属する技術分野】

本発明は、復号方法および復号装置、並びにプログラムに関し、特に、低密度パリティ検査符号による符号化が施された符号の復号を行う復号方法および復号装置、並びにプログラムに関する。



## 【 0 0 0 2 】

## 【従来の技術】

近年、例えば、移動体通信や深宇宙通信といった通信分野、及び地上波又は衛星デジタル放送といった放送分野の研究が著しく進められているが、それに伴い、誤り訂正符号化及び復号の効率化を目的として符号理論に関する研究も盛んに行われている。

## 【 0 0 0 3 】

符号性能の理論的限界としては、いわゆるシャノン(C. E. Shannon)の通信路符号化定理によって与えられるシャノン限界が知られている。符号理論に関する研究は、このシャノン限界に近い性能を示す符号を開発することを目的として行われている。近年では、シャノン限界に近い性能を示す符号化方法として、例えば、並列連接畳み込み符号(PCCC(Parallel Concatenated Convolutional Codes))や、縦列連接畳み込み符号(SCCC(Serially Concatenated Convolutional Codes))といった、いわゆるターボ符号化(Turbo coding)と呼ばれる手法が開発されている。また、これらのターボ符号が開発される一方で、古くから知られる符号化方法である低密度パリティ検査符号(Low Density Parity Check codes) (以下、LDPC符号という) が脚光を浴びつつある。

## 【 0 0 0 4 】

LDPC符号は、R. G. Gallagerによる「R. G. Gallager, "Low Density Parity Check Codes", Cambridge, Massachusetts: M. I. T. Press, 1963」において最初に提案されたものであり、その後、「D. J. C. MacKay, "Good error correcting codes based on very sparse matrices", Submitted to IEEE Trans. Inf. Theory, IT-45, pp. 399-431, 1999」や、「M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi and D. A. Spielman, "Analysis of low density codes and improved designs using irregular graphs", in Proceedings of ACM Symposium on Theory of Computing, pp. 249-258, 1998」等において再注目されるに至ったものである。

## 【 0 0 0 5 】

LDPC符号は、近年の研究により、ターボ符号等と同様に、符号長を長くしてい

くにしたがって、シャノン限界に近い性能が得られることがわかりつつある。また、LDPC符号は、最小距離が符号長に比例するという性質があることから、その特徴として、ブロック誤り確率特性がよく、さらに、ターボ符号等の復号特性において観測される、いわゆるエラーフロア現象が殆ど生じないことも利点として挙げられる。

#### 【0006】

以下、このようなLDPC符号について具体的に説明する。なお、LDPC符号は、線形符号であり、必ずしも2元である必要はないが、ここでは、2元であるものとして説明する。

#### 【0007】

LDPC符号は、そのLDPC符号を定義する検査行列 (parity check matrix) が疎なものであることを最大の特徴とするものである。ここで、疎な行列とは、行列のコンポーネントの“1”の個数が非常に少なく構成されるものであり、疎な検査行列をHで表すものとする、そのような検査行列としては、例えば、図1に示すように、各行のハミング重み (“1”の数) (weight) が“3”であり、且つ、各列のハミング重みが“6”であるもの等がある。

#### 【0008】

このように、各行及び各列のハミング重みが一定である検査行列Hによって定義されるLDPC符号は、レギュラーLDPC符号と称される。一方、各行及び各列のハミング重みが一定でない検査行列Hによって定義されるLDPC符号は、イレギュラーLDPC符号と称される。

#### 【0009】

このようなLDPC符号による符号化は、検査行列Hに基づいて生成行列Gを生成し、この生成行列Gを2元の情報メッセージに対して乗算することによって符号語を生成することで実現される。具体的には、LDPC符号による符号化を行う符号化装置は、まず、検査行列Hの転置行列 $H^T$ との間に、式 $GH^T=0$ が成立する生成行列Gを算出する。ここで、生成行列Gが、 $k \times n$ 行列である場合には、符号化装置は、生成行列Gに対してkビットからなる情報メッセージ (ベクトルu) を乗算し、nビットからなる符号語c ( $= uG$ ) を生成する。この符号化装置によって生成さ

れた符号語は、値が"0"の符号ビットが"1"に、値が"1"の符号ビットが"1"にといったようにマッピングされて送信され、所定の通信路を介して受信側において受信されることになる。

#### 【0010】

一方、LDPC符号の復号は、Gallagerが確率復号(Probabilistic Decoding)と称して提案したアルゴリズムであって、バリアブルノード(variable node (メッセージノード(message node)ともいう。))と、チェックノード(check node)とからなる、いわゆるタナグラフ(Tanner graph)上での確率伝播(belief propagation)によるメッセージ・パッシング・アルゴリズムによって行うことが可能である。ここで、以下、適宜、バリアブルノードとチェックノードを、単に、ノードともいう。

#### 【0011】

しかしながら、確率復号においては、各ノード間で受け渡されるメッセージが実数値であることから、解析的に解くためには、連続した値をとるメッセージの確率分布そのものを追跡する必要がある、非常に困難を伴う解析を必要とすることになる。そこで、Gallagerは、LDPC符号の復号アルゴリズムとして、アルゴリズムA又はアルゴリズムBを提案している。

#### 【0012】

LDPC符号の復号は、一般的には、図2に示すような手順にしたがって行われる。なお、ここでは、受信値(受信した符号系列)を $U_0(u_{0i})$ とし、チェックノードから出力されるメッセージを $u_j$ とし、バリアブルノードから出力されるメッセージを $v_i$ とする。また、ここでは、メッセージとは、値の"0"らしさを、いわゆる対数尤度比(log likelihood ratio)で表現した実数値である。

#### 【0013】

まず、LDPC符号の復号においては、図2に示すように、ステップS11において、受信値 $U_0(u_{0i})$ が受信され、メッセージ $u_j$ が"0"に初期化されるとともに、繰り返し処理のカウンタとしての整数をとる変数 $k$ が"0"に初期化され、ステップS12に進む。ステップS12において、受信値 $U_0(u_{0i})$ に基づいて、式(1)に示す演算(バリアブルノードの演算)を行うことによってメッセージ $v_i$ が求めら

れ、さらに、このメッセージ $v_i$ に基づいて、式(2)に示す演算(チェックノードの演算)を行うことによってメッセージ $u_j$ が求められる。

【0014】

【数1】

$$v_i = u_{0i} + \sum_{j=1}^{d_v-1} u_j \quad \dots (1)$$

【0015】

【数2】

$$\tanh\left(\frac{u_j}{2}\right) = \prod_{i=1}^{d_c-1} \tanh\left(\frac{v_i}{2}\right) \quad \dots (2)$$

【0016】

ここで、式(1)と式(2)における $d_v$ と $d_c$ は、それぞれ、検査行列 $H$ の縦方向(列)と横方向(行)の"1"の個数を示す任意に選択可能とされるパラメータであり、例えば、(3,6)符号の場合には、 $d_v=3$ 、 $d_c=6$ となる。

【0017】

なお、式(1)または(2)の演算においては、それぞれ、メッセージを出力しようとする枝(edge)(バリアブルノードとチェックノードとを結ぶ線)から入力されたメッセージを、和または積演算のパラメータとしては用いないことから、和または積演算の範囲が、1乃至 $d_v-1$ または1乃至 $d_c-1$ となっている。また、式(2)に示す演算は、実際には、2入力 $v_1$ 、 $v_2$ に対する1出力で定義される式(3)に示す関数 $R(v_1, v_2)$ のテーブルを予め作成しておき、これを式(4)に示すように連続的(再帰的)に用いることによって行われる。

【0018】

【数3】

$$x = 2 \tanh^{-1} \{ \tanh(v_1/2) \tanh(v_2/2) \} = R(v_1, v_2) \quad \dots (3)$$

【0019】

【数4】

$$u_j = R(v_1, R(v_2, R(v_3, \dots R(v_{d_c-2}, v_{d_c-1}))) \dots (4)$$

【0020】

ステップS12では、さらに、変数kが"1"だけインクリメントされ、ステップS13に進む。ステップS13では、変数kが所定の繰り返し復号回数Nよりも大きいかなんかが判定される。ステップS13において、変数kがNよりも大きくないと判定された場合、ステップS12に戻り、以下、同様の処理が繰り返される。

【0021】

また、ステップS13において、変数kがNよりも大きいと判定された場合、ステップS14に進み、式(5)に示す演算を行うことによって最終的に出力する復号結果としてのメッセージ $v_i$ が求められて出力され、LDPC符号の復号処理が終了する。

【0022】

【数5】

$$v_i = u_{0i} + \sum_{j=1}^{d_v} u_j \dots (5)$$

【0023】

ここで、式(5)の演算は、式(1)の演算とは異なり、バリエブルノードに接続している全ての枝からの入力メッセージを用いて行われる。

【0024】

このようなLDPC符号の復号は、例えば(3,6)符号の場合には、図3に示すように、各ノード間でメッセージの授受が行われる。なお、図3における"="で示すノード(バリエブルノード)では、式(1)に示した演算が行われ、"+"で示すノード(チェックノード)では、式(2)に示した演算が行われる。特に、アルゴリズムAにおいては、メッセージを2元化し、"+"で示すノードにて、 $d_c-1$ 個の入力メッセージの排他的論理和演算を行い、"="で示すノードにて、受信値Rに

対して、 $d_v-1$ 個の入力メッセージが全て異なるビット値であった場合には、符号を反転して出力する。

#### 【0025】

また、一方で、近年、LDPC符号の復号の実装法に関する研究も行われている。実装方法について述べる前に、まず、LDPC符号の復号を模式化して説明する。

#### 【0026】

図4は、(3,6)LDPC符号(符号化率1/2、符号長12)の検査行列(parity check matrix)の例である。LDPC符号の検査行列は、図5のように、タナーグラフを用いて書き表すことができる。ここで、図5において、“+”で表わされるのが、チェックノードであり、“=”で表わされるのが、バリエブルノードである。チェックノードとバリエブルノードは、それぞれ、検査行列の行と列に対応する。チェックノードとバリエブルノードとの間の結線は、枝(edge)であり、検査行列の“1”に相当する。即ち、検査行列の第 $j$ 行第 $i$ 列のコンポーネントが1である場合には、図5において、上から $i$ 番目のバリエブルノード(“=”のノード)と、上から $j$ 番目のチェックノード(“+”のノード)とが、枝により接続される。枝は、バリエブルノードに対応する符号ビットが、チェックノードに対応する拘束条件を持つことを表わす。なお、図5は、図4の検査行列のタナーグラフとなっている。

#### 【0027】

LDPC符号の復号方法であるサンプロダクトアルゴリズム(Sum Product Algorithm)では、バリエブルノードの演算とチェックノードの演算とが繰り返し行われる。

#### 【0028】

バリエブルノードでは、図6のように、式(1)の演算(バリエブルノード演算)を行う。すなわち、図6において、計算しようとしている枝に対応するメッセージ $v_i$ は、バリエブルノードに繋がっている残りの枝からのメッセージ $u_1$ および $u_2$ と、受信情報 $u_{0i}$ を用いて計算される。他の枝に対応するメッセージも同様に計算される。

#### 【0029】

次に、チェックノードの演算について説明する前に、式(2)を、式 $a \times b = \exp\{\ln(|a|) + \ln(|b|)\} \times \text{sign}(a) \times \text{sign}(b)$ の関係を用いて、式(6)のように書き直す。但し、 $\text{sign}(x)$ は、 $x \geq 0$ のとき1であり、 $x < 0$ のとき-1である。

【0030】

【数6】

$$\begin{aligned}
 u_j &= 2 \tanh^{-1} \left( \prod_{i=1}^{d_c-1} \tanh \left( \frac{v_i}{2} \right) \right) \\
 &= 2 \tanh^{-1} \left[ \exp \left\{ \sum_{i=1}^{d_c-1} \ln \left( \left| \tanh \left( \frac{v_i}{2} \right) \right| \right) \right\} \times \prod_{i=1}^{d_c-1} \text{sign} \left( \tanh \left( \frac{v_i}{2} \right) \right) \right] \\
 &= 2 \tanh^{-1} \left[ \exp \left\{ - \left( \sum_{i=1}^{d_c-1} - \ln \left( \tanh \left( \frac{|v_i|}{2} \right) \right) \right) \right\} \times \prod_{i=1}^{d_c-1} \text{sign}(v_i) \right] \\
 &\quad \dots (6)
 \end{aligned}$$

【0031】

更に、 $x \geq 0$ において、 $\phi(x) = \ln(\tanh(x/2))$ と定義すると、 $\phi^{-1}(x) = 2 \tanh^{-1}(e^{-x})$ であるから、式(6)は、式(7)のように書くことができる。

【0032】

【数7】

$$u_j = \phi^{-1} \left( \sum_{i=1}^{d_c-1} \phi(|v_i|) \right) \times \prod_{i=1}^{d_c-1} \text{sign}(v_i) \quad \dots (7)$$

【0033】

チェックノードでは、図7のように、式(7)の演算(チェックノード演算)を行う。すなわち、図7において、計算しようとしている枝に対応するメッセージ $u_j$ は、チェックノードに繋がっている残りの枝からのメッセージ $v_1, v_2, v_3, v_4, v_5$ を用いて計算される。他の枝に対応するメッセージも同様に計算される。

【0034】

なお、関数 $\phi(x)$ は、 $\phi(x) = \ln((e^x + 1)/(e^x - 1))$ とも表すことができ、 $x > 0$ において、 $\phi(x) = \phi^{-1}(x)$ である。関数 $\phi(x)$ および $\phi^{-1}(x)$ をハードウェアに実装す

る際には、LUT(Look Up Table)を用いて実装される場合があるが、両者共に同一のLUTとなる。

#### 【0035】

サムプロダクトアルゴリズムをハードウェアに実装する場合、式(1)で表わされるバリアブルノード演算および式(7)で表わされるチェックノード演算とを、適度な回路規模と動作周波数で繰り返し行うことが必要である。

#### 【0036】

復号装置の実装の例として、まず、単純に各ノードの演算を一つずつ順次行うことによって復号を行う場合(full serial decoding)の実装法について説明する。

#### 【0037】

なお、ここでは、例えば、図8の、30(行)×90(列)の検査行列で表現される符号(符号化率2/3、符号長90)を復号することとする。図8の検査行列の1の数は269であり、従って、そのタナグラフでは、枝の数は269個となる。ここで、図8の検査行列では(後述する図15乃至図17においても同様)、0を、“.”で表現している。

#### 【0038】

図9は、LDPC符号の1回復号を行う復号装置の構成例を示している。

#### 【0039】

図9の復号装置では、その動作する1クロック(clock)ごとに、1つの枝に対応するメッセージが計算される。

#### 【0040】

即ち、図9の復号装置は、1つの受信用メモリ104、2つの枝用メモリ100および102、1つのチェックノード計算器101、1つのバリアブルノード計算器103からなる。

#### 【0041】

図9の復号装置では、枝用メモリ100または102からメッセージデータが一つずつ読み出され、そのメッセージデータを用いて、所望の枝に対応するメッセージデータが計算される。そして、その計算によって求められたメッセージデ



ータが一つずつ後段の枝用メモリ102または100に格納されていく。繰り返し復号を行う際には、この1回復号を行う図9の復号装置を複数個縦列に接続するか、もしくは図9の復号装置を繰り返し用いることによって、繰り返し復号を実現する。なお、ここでは、例えば、図9の復号装置が複数個接続されているものとする。

#### 【0042】

枝用メモリ100は、前段の復号装置（図示せず）のバリアブルノード計算器103から供給される出力メッセージD100を、後段のチェックノード計算器101が読み出す順番に格納していく。そして、枝用メモリ100は、チェックノード計算のフェーズでは、メッセージD100を、格納してある順番通りに、メッセージ出力D101として、チェックノード計算器101に供給する。チェックノード計算器101は、枝用メモリ100から供給されるメッセージD101を用いて、式（7）に従って演算を行い、その演算によって求められたメッセージD102を、後段の枝用メモリ102に供給する。

#### 【0043】

ここで、図10は、チェックノード計算を一つずつ行う図9のチェックノード計算器101の構成例を示している。

#### 【0044】

図10のチェックノード計算器101では、枝用メモリ100から供給される、検査行列の各列に対応するバリアブルノードからのメッセージ $v_i$ を一つずつ読み込み、式（7）における $\phi(|v_i|)$ の演算をLUTによって行う。さらに、検査行列の1行に亘る各列に対応するバリアブルノードからのメッセージ $v_i$ から求められた $\phi(|v_i|)$ が積算され、これにより、全ての枝からのメッセージ $v_i$ から求められた $\phi(|v_i|)$ の積算値が求められる。その後、その積算値から、メッセージ $u_j$ を求めたい枝から求められてFIFO（FIFOメモリ）で遅延された $\phi(|v_i|)$ が減算され、これにより、メッセージ $u_j$ を求めたい枝について、式（7）における $\sum \phi(|v_i|)$ が求められる。即ち、チェックノードへの枝すべてからのメッセージの和から、メッセージ $u_j$ を求めたい枝からのメッセージを減算することで、メッセージ $u_j$ を求めたい枝へのメッセージが求められる。さらに、LUTによって、式（7）に

おける  $\phi^{-1}(\sum \phi(|v_i|))$  の演算が行われる。同時に、メッセージ  $u_j$  の符号ビット、即ち、式 (7) における  $\Pi \text{sign}(v_i)$  も、EXOR回路を用いて同様に計算される。以上のようにして、式 (7) の演算が行われ、メッセージ  $u_j$  が求められる。

#### 【0045】

なお、図10では、各メッセージが符号ビットを合わせて合計6ビット(bit)に量子化されているものとして、チェックノード計算器101を表している。また、ここで処理の対象としている図8の検査行列の行の重み(row weight)の最大は9であるため、即ち、チェックノードに供給されるメッセージの最大数は9であるため、チェックノード計算器101は、9個のメッセージ ( $\phi(|v_i|)$ ) を遅延させるFIFO(First In First Out)を持っている。

#### 【0046】

図9に戻り、枝用メモリ102は、前段のチェックノード計算器101から供給される出力メッセージD102を、後段のバリアブルノード計算器103が読み出す順番に格納していく。そして、枝用メモリ102は、バリアブルノード計算のフェーズでは、メッセージ出力D102を、格納してある順番通りに、メッセージ出力D103として、バリアブルノード計算器103に供給する。

#### 【0047】

バリアブルノード計算器103は、枝用メモリ102から供給されるメッセージD103と受信用メモリ104から供給される受信データ (LDPC符号の受信値) D104を用いて式 (1) に従って演算を行い、その演算の結果得られるメッセージD105を、図示せぬ後段の復号装置の枝用メモリ100に供給する。

#### 【0048】

ここで、図11は、バリアブルノード計算を一つずつ行う図9のバリアブルノード計算器103の構成例を示している。

#### 【0049】

図11のバリアブルノード計算器103では、枝用メモリ102から供給される、検査行列の各行に対応するチェックノードからのメッセージ  $u_j$  を一つずつ読み込み、検査行列の1列に亘る各行に対応するチェックノードからのメッセージを積算して、その積算値を求める。その後、その積算値から、メッセージ  $v_i$  を求

めたい枝から供給されてFIFOで遅延されたメッセージが減算される。さらに、その結果得られる減算値から、受信値 $u_{0i}$ を加算することで、式(1)の演算が行われ、これにより、メッセージ $v_i$ が求められる。即ち、バリエブルノードへの枝すべてからのメッセージの和から、メッセージ $v_i$ を求めたい枝からのメッセージを減算することで、メッセージ $v_i$ を求めたい枝へのメッセージが求められる。

#### 【0050】

図11においても、図10における場合と同様に、各メッセージが符号ビットを合わせて合計6ビットに量子化されているものとして、バリエブルノード計算器103を表している。また、ここで処理の対象としている図8の検査行列においては、列の重み(column weight)の最大値が5であるため、バリエブル計算器103は、5個のメッセージを遅延させるFIFOを持っており、列の重みが5未満の列のメッセージを計算するときには、FIFOにおける遅延量が、その列の重みの値に減らされる。

#### 【0051】

再び、図9に戻り、復号装置には、検査行列の重みにしたがって、図示しない制御信号が与えられる。そして、図9の復号装置によれば、枝用メモリ100および102、並びにチェックノード計算器101およびバリエブルノード計算器103のFIFOの容量さえ足りれば、制御信号のみを変えることで様々な符号を復号することができる。

#### 【0052】

なお、図示しないが、図9の復号装置において、復号の最終段においては、式(1)のバリエブルノード演算の代わりに、式(5)の演算が行われ、その演算結果が、最終的な復号結果として出力される。

#### 【0053】

図9の復号装置を繰り返し用いて、LDPC符号を復号する場合には、チェックノード演算とバリエブルノード演算とが交互に行われるため、269の枝を有する図8の検査行列を用いた1回の復号に、 $269 \times 2 = 538$ クロック(clock)を必要とする。従って、例えば、50回の繰り返し復号を行うためには、符号長である90個の符号情報(受信値)を受信する間に、 $538 \times 50 = 26900$ クロック動作すること

が必要であり、受信周波数の約 300 ( $\div 26900/90$ ) 倍の高速動作が必要になる。この場合、受信周波数が数十MHzであるとする、GHz以上の速度での動作を要求されることになり、実装は容易ではない。

#### 【0054】

また、図9の復号装置を、例えば、50台接続して、LDPC符号を復号する場合には、1フレーム(frame)目がバリエブルノード演算を行っている間に、2フレーム目はチェックノード演算を行い、3フレーム目は前段のバリエブルノード演算を行う、というように、複数のバリエブルノード演算とチェックノード演算とを同時に行うことができる。この場合、90個の符号情報を受信する間に、269個の枝を計算すればよいので、復号装置は、受信周波数の約3 ( $\div 269/90$ ) 倍の周波数で動作すればよいことになり、十分に実現可能である。しかしながら、この場合、回路規模が、単純には、図9の復号装置の50倍になる。

#### 【0055】

次に、全ノードの演算を同時に行うことによって復号を行う場合(full parallel decoding)の復号装置の実装法について説明する。

#### 【0056】

この実装法については、例えば、非特許文献1に記載されている。

#### 【0057】

図12は、図8の検査行列で表現される符号(符号化率2/3、符号長90)を復号する復号装置の一例の構成を示している。

#### 【0058】

図12の復号装置では、枝用メモリ202または206から、269個ある枝に対応するメッセージデータを全て同時に読み出し、そのメッセージデータを用いて、269個の枝に対応する新たなメッセージデータを演算する。さらに、その演算の結果求められた新たなメッセージデータが全て同時に後段の枝用メモリ206または202に格納されていく。そして、図12の復号装置を繰り返し用いることで繰り返し復号が実現される。

#### 【0059】

図12において、復号装置は、1つの受信用メモリ205、2つの枝入れ替え

装置 200 および 203、2つの枝用メモリ 202 および 206、30個のチェックノード計算器 201<sub>1</sub>乃至201<sub>30</sub>、90個のバリアブルノード計算器 204<sub>1</sub>乃至204<sub>90</sub>からなる。以下、各部について詳細に説明する。

#### 【0060】

枝用メモリ 206 は、前段のバリアブルノード計算器 204<sub>1</sub>乃至204<sub>90</sub>からの出力メッセージ D206<sub>1</sub>乃至D206<sub>90</sub>を全て同時に格納し、次の時刻（次のクロックのタイミング）に、メッセージ D206<sub>1</sub>乃至D206<sub>90</sub>を、メッセージ D207<sub>1</sub>乃至D207<sub>90</sub>として読み出し、次段の枝入れ替え装置 200 に、メッセージ D200 (D200<sub>1</sub>乃至D200<sub>90</sub>)として供給する。枝入れ替え装置 200 は、枝用メモリ 206 から供給されたメッセージ D200<sub>1</sub>乃至D200<sub>90</sub>の順番を、図 8 の検査行列に従って並び替え（入れ替え）、チェックノード計算器 201<sub>1</sub>乃至201<sub>30</sub>に、それぞれ必要なメッセージ D201<sub>1</sub>乃至D201<sub>30</sub>を供給する。

#### 【0061】

チェックノード計算器 201<sub>1</sub>乃至201<sub>30</sub>は、枝入れ替え装置 200 から供給されるメッセージ D201<sub>1</sub>乃至D201<sub>30</sub>を用いて式 (7) に従って演算を行い、その演算の結果得られるメッセージ D202<sub>1</sub>乃至D202<sub>30</sub>を、枝用メモリ 202 に供給する。

#### 【0062】

ここで、図 13 は、チェックノード演算を同時に行う図 12 のチェックノード計算器 201<sub>m</sub> (m=1, 2, ..., 30) の構成例を示している。

#### 【0063】

図 13 のチェックノード計算器 201<sub>m</sub>では、図 10 のチェックノード計算器 101 と同様にして、式 (7) のチェックノード演算が行われるが、そのチェックノード演算が、すべての枝について同時に行われる。

#### 【0064】

即ち、図 13 のチェックノード計算器 201<sub>m</sub>では、枝入れ替え装置 200 から供給される図 8 の検査行列の各列に対応するバリアブルノードからのメッセージが全て同時に読み込まれ、式 (7) における  $\phi(|v_i|)$  の演算が LUT によって行われる。さらに、検査行列の 1 行に亘る各列に対応するバリアブルノードからの

メッセージ $v_i$ から求められた $\phi(|v_i|)$ が積算され、これにより、全ての枝からのメッセージ $v_i$ から求められた $\phi(|v_i|)$ の積算値が求められる。その後、その積算値から、メッセージ $u_j$ を求めたい枝から求められた $\phi(|v_i|)$ が減算され、これにより、メッセージ $u_j$ を求めたい枝について、式(7)における $\Sigma \phi(|v_i|)$ が求められる。即ち、チェックノードへの枝すべてからのメッセージの和から、メッセージ $u_j$ を求めたい枝からのメッセージを減算することで、メッセージ $u_j$ を求めたい枝へのメッセージが求められる。さらに、LUTによって、式(7)における $\phi^{-1}(\Sigma \phi(|v_i|))$ の演算が行われる。同時に、メッセージ $u_j$ の符号ビット、即ち、式(7)における $\Pi \text{sign}(v_i)$ も、EXOR回路を用いて同様に計算される。以上のようにして、式(7)の演算が行われ、メッセージ $u_j$ が求められる。

#### 【0065】

なお、図13では、各メッセージが符号ビットを合わせて合計6ビットに量子化されているものとして、チェックノード計算器201<sub>m</sub>を表している。また、図13の回路は一つのチェックノードに相当する。ここで処理の対象としている図8の検査行列については、その行数である30行のチェックノードが存在するから、図12の復号装置は、図13に示したようなチェックノード計算器201<sub>m</sub>を30個有している。

#### 【0066】

ここで、図13のチェックノード計算器201<sub>m</sub>では、9個のメッセージを同時に計算することができる。そして、ここで処理の対象としている図8の検査行列の行の重みは、第1行が8で、第2乃至第30行が9であるため、即ち、チェックノードに供給されるメッセージの数が、8のケースが1つと、9のケースが29あるため、チェックノード計算器201<sub>1</sub>は、図13の回路と同様の8つのメッセージを同時に計算することができる回路構成となっており、チェックノード計算器201<sub>2</sub>乃至201<sub>30</sub>は、図13の回路と同一構成となっている。

#### 【0067】

図12に戻り、枝用メモリ202は、前段のチェックノード計算器201<sub>1</sub>乃至201<sub>30</sub>から供給される出力メッセージD202<sub>1</sub>乃至D202<sub>30</sub>を全て同時に格納し、次の時刻に、そのすべてのメッセージD202<sub>1</sub>乃至D202<sub>30</sub>を、出力メッセージD20

$3_1$ 乃至 $D203_{30}$ として、次段の枝入れ替え装置 203 に供給する。

#### 【0068】

枝入れ替え装置 203 は、枝用メモリ 202 から供給されたメッセージ $D203_1$ 乃至 $D203_{30}$ の順番を図 8 の検査行列に従って並び替え、バリエブルノード計算器 204<sub>1</sub>乃至 204<sub>90</sub>に、それぞれ必要なメッセージ $D204_1$ 乃至 $D204_{90}$ を供給する。

#### 【0069】

バリエブルノード計算器 204<sub>1</sub>乃至 204<sub>90</sub>は、枝入れ替え装置 203 から供給されるメッセージ $D204_1$ 乃至 $D204_{90}$ と、受信用メモリ 205 から供給される受信データ（受信値） $D205_1$ 乃至 $D205_{90}$ を用いて式（1）に従って演算を行い、その演算の結果得られるメッセージ $D206_1$ 乃至 $D206_{90}$ を、次段の枝用メモリ 206 に供給する。

#### 【0070】

ここで、図 14 は、バリエブルノード演算を同時に行う図 12 のバリエブルノード計算器 204<sub>p</sub>（ $p=1, 2, \dots, 90$ ）の構成例を示している。

#### 【0071】

図 14 のバリエブルノード計算器 204<sub>p</sub>では、図 11 のバリエブルノード計算器 103 と同様にして、式（7）のチェックノード演算が行われるが、そのチェックノード演算が、すべての枝について同時に行われる。

#### 【0072】

即ち、図 14 のバリエブルノード計算器 204<sub>p</sub>では、枝入れ替え装置 203 から供給される、検査行列の各行に対応するチェックノードからのメッセージ $u_j$ が全て同時に読み込まれ、検査行列の 1 列に亘る各行に対応するチェックノードからのメッセージが積算されて、その積算値が求められる。その後、その積算値から、メッセージ $v_i$ を求めたい枝から供給されたメッセージが減算され、その結果得られる減算値から、受信値 $u_{0i}$ を加算することで、式（1）の演算が行われ、これにより、メッセージ $v_i$ が求められる。即ち、バリエブルノードへの枝すべてのメッセージの和から、メッセージ $v_i$ を求めたい枝からのメッセージを減算することで、メッセージ $v_i$ を求めたい枝へのメッセージが求められる。

## 【0073】

なお、図14では、各メッセージが符号ビットを合わせて合計6ビットに量子化されているものとして、バリアブルノード計算器204<sub>p</sub>を表している。また、図14の回路は一つのバリアブルノードに相当する。ここで処理の対象としている図8の検査行列については、その列数である90列のバリアブルノードが存在するから、図12の復号装置は、図14に示したようなバリアブルノード計算器204<sub>p</sub>を90個有している。

## 【0074】

ここで、図14のバリアブルノード計算器204<sub>p</sub>では、5個のメッセージを同時に計算することができる。そして、ここで処理の対象としている図8の検査行列は、重みが5, 3, 2, 1の列が、それぞれ、15列、45列、29列、1列あるので、バリアブルノード計算器204<sub>1</sub>乃至204<sub>90</sub>のうちの15個は、図14の回路と同一構成となっており、残りの45, 29, 1個は、図14の回路と同様の3, 2, 1つのメッセージを同時にそれぞれ計算することができる回路構成となっている。

## 【0075】

なお、図示しないが、図12の復号装置においても、図9における場合と同様に、復号の最終段においては、式(1)のバリアブルノード演算の代わりに、式(5)の演算が行われ、その演算結果が復号結果として出力される。

## 【0076】

図12の復号装置によれば、269個ある枝に対応するメッセージすべてを1クロックで同時に計算することができる。

## 【0077】

図12の復号装置を繰り返し用いて復号する場合には、チェックノード演算とバリアブルノード演算とを交互に行い、1回の復号を2クロックで行うことができる。従って、例えば、50回の復号を行うためには、90個の符号情報を受信する間に  $2 \times 50 = 100$  クロック動作すれば良いことになり、ほぼ受信周波数と同一の動作周波数でよいことになる。一般的に、LDPC符号は、符号長が数千から数万と大きいことから、図12の復号装置を用いれば、復号回数を極めて多くする



ことができ、誤り訂正性能の向上が期待できる。

【0078】

しかしながら、図12の復号装置は、タナーグラフのすべての枝に対応するメッセージの演算を、並列で行うため、回路規模が、符号長に比例して大きくなる。また、図12の復号装置を、ある符号長の、ある符号化率の、ある検査行列を持つLDPC符号の復号を行う装置として構成した場合、その復号装置において、他の符号長や、他の符号化率、他の検査行列を持つLDPC符号の復号を行うことは困難となる。即ち、図12の復号装置は、図9の復号装置のように、制御信号を変えるだけでは、様々な符号を復号することはできず、符号依存性が高い。

【0079】

図9および図12の復号装置の他に、一つでも全てでもなく、4つずつのメッセージの計算を同時に行う実装法について、例えば、非特許文献2に述べられているが、この場合、メモリの異なるアドレスからの同時読み出し、もしくは同時書き込みを避けることが一般的には容易でなく、メモリアクセス制御が困難であるという問題がある。

【0080】

また、サムプロダクトアルゴリズムを近似して実装する方法なども提案されているが、この方法では、性能の劣化を招いてしまう。

【0081】

【非特許文献1】

C. Howland and A. Blanksby, "Parallel Decoding Architectures for Low Density Parity Check Codes", Symposium on Circuits and Systems, 2001

【0082】

【非特許文献2】

E. Yeo, P. Pakzad, B. Nikolic and V. Anantharam, "VLSI Architectures for iterative Decoders in Magnetic Recording Channels", IEEE Transactions on Magnetics, Vol. 37, No. 2, March 2001

【0083】

【発明が解決しようとする課題】

図 1 5 は、符号長 90、符号化率  $2/3$  の LDPC 符号の検査行列の一例である。この検査行列で表される LDPC 符号の復号装置を実装することを考える場合、枝に対応するメッセージを一つずつ計算する復号装置、もしくは枝に対応するメッセージを全て同時に計算する復号装置の設計自体は難しいことではない。

#### 【 0 0 8 4 】

しかしながら、その復号装置の実現は、回路規模や動作速度の面から見て、容易ではない。

#### 【 0 0 8 5 】

また、図 1 5 の検査行列で表される符号を、ある数  $P$  個の枝を同時に計算する復号装置を用いて復号する場合、枝データ（枝に対応するメッセージ）を格納するメモリにおいては、各行または各列毎に異なる位置（アドレス）からの読み出しまたは書き込みのアクセスが必要になるため、各行または各列毎に別々の FIFO を用いることが必要になる。更に、メッセージについては、チェックノード演算で計算された順序と、次のバリアブルノード演算で使われる順番が入れ替わることもあり、メッセージを格納するメモリを、単純に FIFO で実現することも容易ではない。

#### 【 0 0 8 6 】

本発明は、このような状況に鑑みてなされたものであり、ロジック、メモリ共に回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行うことができるようにするものである。

#### 【 0 0 8 7 】

##### 【課題を解決するための手段】

本発明の復号方法は、元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、LDPC 符号を復号する復号ステップを備えることを特徴とする。

#### 【 0 0 8 8 】

本発明の復号装置は、元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、LDPC 符号を復号する復号手段を備えることを特徴とする。

## 【0089】

本発明のプログラムは、元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、LDPC符号を復号する復号ステップを備えることを特徴とする。

## 【0090】

本発明においては、元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、LDPC符号が復号される。

## 【0091】

## 【発明の実施の形態】

以下に本発明の実施の形態を説明するが、請求項に記載の構成要件と、発明の実施の形態における具体例との対応関係を例示すると、次のようになる。この記載は、請求項に記載されている発明をサポートする具体例が、発明の実施の形態に記載されていることを確認するためのものである。従って、発明の実施の形態中には記載されているが、構成要件に対応するものとして、ここには記載されていない具体例があったとしても、そのことは、その具体例が、その構成要件に対応するものではないことを意味するものではない。逆に、具体例が構成要件に対応するものとしてここに記載されていたとしても、そのことは、その具体例が、その構成要件以外の構成要件には対応しないものであることを意味するものでもない。

## 【0092】

さらに、この記載は、発明の実施の形態に記載されている具体例に対応する発明が、請求項に全て記載されていることを意味するものではない。換言すれば、この記載は、発明の実施の形態に記載されている具体例に対応する発明であって、この出願の請求項には記載されていない発明の存在、すなわち、将来、分割出願されたり、補正により追加される発明の存在を否定するものではない。

## 【0093】

請求項4に記載の復号装置は、

LDPC(Low Density Parity Check)符号の復号装置であって、

元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得ら

れる変換検査行列を用いて、前記LDPC符号を復号する復号手段（例えば、図18の枝データ格納メモリ300および304、チェックノード計算部302、サイクリックシフト回路303および308、バリアブルノード計算部307、並びに復号語計算部309）を備えることを特徴とする。

#### 【0094】

請求項6に記載の復号装置は、

前記復号手段は、

前記LDPC符号の復号のためのP個のチェックノードの演算を同時に行うチェックノード計算手段（例えば、図18のチェックノード計算部302）と、

前記LDPC符号の復号のためのP個のバリアブルノードの演算を同時に行うバリアブルノード計算手段（例えば、図18のバリアブルノード計算部307）と

を有する

ことを特徴とする。

#### 【0095】

請求項7に記載の復号装置は、

前記チェックノード計算手段は、チェックノードの演算を行うP個のチェックノード計算器（例えば、図18のチェックノード計算器302<sub>1</sub>乃至302<sub>5</sub>）を有し、

前記バリアブルノード計算手段は、バリアブルノードの演算を行うP個のバリアブルノード計算器（例えば、図18のバリアブルノード計算器307<sub>1</sub>乃至307<sub>5</sub>）を有する

ことを特徴とする。

#### 【0096】

請求項8に記載の復号装置は、

前記復号手段は、前記P個のチェックノードの演算、または前記P個のバリアブルノードの演算の結果得られるP個の枝に対応するメッセージデータを同時に読み書きするメッセージ記憶手段（例えば、図18の枝データ格納メモリ300または304）をさらに有する

ことを特徴とする。

**【0097】**

請求項14に記載の復号装置は、

前記復号手段は、受信情報を格納するとともに、P個の前記受信情報を同時に読み出す受信情報記憶手段（例えば、図18の受信データ用メモリ306）をさらに有する

ことを特徴とする。

**【0098】**

請求項16に記載の復号装置は、

前記復号手段は、前記P個のチェックノードの演算、または前記P個のバリエーションノードの演算の結果得られるメッセージをサイクリックシフトするサイクリックシフト手段（例えば、図18のサイクリックシフト回路303または308）をさらに有する

ことを特徴とする。

**【0099】**

請求項18に記載の復号装置は、

受信した前記LDPC符号の符号系列に対して、前記元の検査行列に対して行われた列置換と同一の列置換を行い、置換符号系列を出力する符号系列置換手段（例えば、図18の受信データ並べ替え部310）をさらに備え、

前記復号手段は、前記変換検査行列と、前記置換符号系列とを用いて、前記符号系列を復号する

ことを特徴とする。

**【0100】**

請求項19に記載の復号装置は、

前記復号手段の出力に対して、前記元の検査行列に対して行われた列置換の逆置換を行い、最終的な復号結果を出力する逆置換手段（例えば、図18の復号データ並べ替え部311）をさらに備える

ことを特徴とする。

**【0101】**

以下、本発明を適用した具体的な実施の形態について、図面を参照しながら詳

細に説明する。

### 【0102】

図16は、図15の検査行列に、式(8)の行置換と、式(9)の列置換を施して得られる検査行列を示している。

### 【0103】

行置換:  $6x+y+1$ 行目  $\rightarrow$   $5y+x+1$ 行目

... (8)

### 【0104】

列置換:  $6s+t+61$ 列目  $\rightarrow$   $5t+s+61$ 列目

... (9)

### 【0105】

但し、式(8)および(9)において、 $x, y, s, t$ は、それぞれ、 $0 \leq x < 5$ ,  $0 \leq y < 6$ ,  $0 \leq s < 5$ ,  $0 \leq t < 6$ の範囲の整数である。

### 【0106】

式(8)の行置換によれば、6で割って余りが1になる1, 7, 13, 19, 25行目を、それぞれ、1, 2, 3, 4, 5行目に、6で割って余りが2になる2, 8, 14, 20, 26行目を、それぞれ、6, 7, 8, 9, 10行目に、という具合に置換が行われる。

### 【0107】

また、式(9)の列置換によれば、61列目以降に対して、6で割って余りが1になる61, 67, 73, 79, 85列目を、それぞれ、61, 62, 63, 64, 65列目に、6で割って余りが2になる62, 68, 74, 80, 86列目を、それぞれ、66, 67, 68, 69, 70列目に、という具合に置換が行われる。

### 【0108】

このようにして、図15の検査行列に対して、行と列の置換を行って得られた行列(matrix)が、図16の検査行列である。

### 【0109】

図16の検査行列(以下、適宜、置換検査行列という)に対して、図15の検

査行列（以下、適宜、元の検査行列という）で表わされる符号（誤りのない元の符号）の符号語の系列に、式（9）と同一の置換を行ったものを乗じると、0ベクトルが出力されることは自明である。即ち、元の検査行列を行列Hで、置換検査行列を行列H'で、元の符号の符号語の系列を行ベクトルcで、行ベクトルcに式（9）の列置換を施して得られる行ベクトルをc'で、それぞれ表すこととすると、検査行列の性質から、 $Hc^T$ （上付のTは転置を表す）は、0ベクトルとなるから、 $H'c'^T$ も、当然、0ベクトルとなる。

#### 【0110】

そして、以上のことから、図16の変換検査行列は、元の符号の符号語の系列cに、式（9）の列置換を行ったものを符号語とする符号c'の検査行列になっている。

#### 【0111】

従って、元の符号によって符号化されたデータを受信して復号する際に、受信した符号系列に、式（9）の列置換を行い、その列置換後の符号系列を、図16の変換検査行列に基づく復号装置を用いて復号し、復号結果の系列に、式（2）の列置換の逆置換を行っても、元の符号の復号装置を用いた場合と復号結果に違いはないため、性能の劣化を招くことはないことになる。即ち、図15の元の検査行列に基づく復号装置は、図16の変換検査行列に基づく復号装置を用いて実現することができる。

#### 【0112】

次に、図17は、 $5 \times 5$ の行列の単位に間隔を空けた、図16の変換検査行列を示している。

#### 【0113】

図17においては、検査行列（変換検査行列）は、 $5 \times 5$ の単位行列、その単位行列の1のうち1個以上が0になった行列（以下、適宜、準単位行列という）、単位行列または準単位行列をサイクリックシフト(cyclic shift)した行列（以下、適宜、シフト行列という）、単位行列、準単位行列、またはシフト行列のうちの2以上の和（以下、適宜、和行列という）、 $5 \times 5$ の0行列の組合わせで表わされている。

## 【0114】

図17の検査行列は、 $5 \times 5$ の単位行列、準単位行列、シフト行列、和行列、0行列で構成されているといふことができる。そこで、検査行列を構成する、これらの $5 \times 5$ の行列を、以下、適宜、構成行列という。

## 【0115】

以上のような $P \times P$ の構成行列で表される検査行列で表される符号の復号には、チェックノードとバリアブルノードの計算を、 $P$ 個同時に行うアーキテクチャ(architecture)を用いることができる。

## 【0116】

図18は、そのような復号装置の一実施の形態の構成例を示すブロック図である。即ち、図18は、図15の元の検査行列に対して、行または列置換を行って得られる図17の変換検査行列を用いて、LDPC符号の復号を行う復号装置の構成例を示している。

## 【0117】

この復号装置は、6つのFIFO300<sub>1</sub>乃至300<sub>6</sub>からなる枝データ格納用メモリ300、FIFO300<sub>1</sub>乃至300<sub>6</sub>を選択するセレクタ301、チェックノード計算部302、2つのサイクリックシフト回路303および308、18個のFIFO304<sub>1</sub>乃至304<sub>18</sub>からなる枝データ格納用メモリ304、FIFO304<sub>1</sub>乃至304<sub>18</sub>を選択するセレクタ305、受信情報を格納する受信データ用メモリ306、バリアブルノード計算部307、復号語計算部309、受信データ並べ替え部310、復号データ並べ替え部311からなる。

## 【0118】

この復号装置の各部について詳細に説明する前に、まず、枝データ格納用メモリ300と304へのデータの格納方法について説明する。

## 【0119】

枝データ格納用メモリ300は、図17の変換検査行列の行数30を構成行列の行数5で除算した数である6つのFIFO300<sub>1</sub>乃至300<sub>6</sub>から構成されている。FIFO300<sub>y</sub> ( $y=1, 2, \dots, 6$ )は、構成行列の行数および列数である5つの枝に対応するメッセージを同時に読み出しもしくは書き込むことができ



るようになっており、その長さ（段数）は、図17の変換検査行列の行方向の1の数（ハミング重み）の最大数である9になっている。

### 【0120】

FIFO300<sub>1</sub>には、図17の検査行列（変換検査行列）の第1行目から第5行目までの1の位置に対応するデータが、各行共に横方向に詰めた形に（0を無視した形で）格納される。すなわち、第j行第i列を、(j,i)と表すこととすると、FIFO300<sub>1</sub>の第1の要素（第1段）には、検査行列の(1,1)から(5,5)の5×5の単位行列の1の位置に対応するデータが格納される。第2の要素には、検査行列の(1,21)から(5,25)のシフト行列（5×5の単位行列を右方向に3つだけサイクリックシフトしたシフト行列）の1の位置に対応するデータが格納される。第3から第8の要素も同様に検査行列と対応づけてデータが格納される。そして、第9の要素には、検査行列の(1,86)から(5,90)のシフト行列（5×5の単位行列のうちの1行目の1を0に置き換えて1つだけ左にサイクリックシフトしたシフト行列）の1の位置に対応するデータが格納される。ここで、検査行列の(1,86)から(5,90)のシフト行列においては、1行目に1がないため、FIFO300<sub>1</sub>の1行目のみ要素数は8、残りの行は要素数が9となる。

### 【0121】

FIFO300<sub>2</sub>には、図17の検査行列の第6行目から第10行目までの1の位置に対応するデータが格納される。すなわち、FIFO300<sub>2</sub>の第1の要素には、検査行列の(6,1)から(10,5)の和行列（5×5の単位行列を右に1つだけサイクリックシフトした第1のシフト行列と、右に2つだけサイクリックシフトした第2のシフト行列の和である和行列）を構成する第1のシフト行列の1の位置に対応するデータが格納される。また、第2の要素には、検査行列の(6,1)から(10,5)の和行列を構成する第2のシフト行列の1の位置に対応するデータが格納される。

### 【0122】

即ち、重みが2以上の構成行列については、その構成行列を、重みが1であるP×Pの単位行列、そのコンポーネントである1のうち1個以上が0になった準単位行列、または単位行列もしくは準単位行列をサイクリックシフトしたシフト行

列のうちの複数の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列の1の位置に対応するデータ（単位行列、準単位行列、またはシフト行列に属する枝に対応するメッセージ）は、同一アドレス（FIFO300<sub>1</sub>乃至300<sub>6</sub>のうちの同一のFIFO）に格納される。

#### 【0123】

以下、第3から第9の要素についても、検査行列に対応づけてデータが格納される。FIFO300<sub>2</sub>は全行共に要素数は9となる。

#### 【0124】

FIFO300<sub>3</sub>乃至300<sub>6</sub>も同様に検査行列に対応づけてデータを格納し、各FIFO300<sub>3</sub>乃至300<sub>6</sub>それぞれの長さは9である。

#### 【0125】

枝データ格納用メモリ304は、検査行列の列数90を、構成行列の列数である5で割った18個のFIFO304<sub>1</sub>乃至304<sub>18</sub>から構成されている。FIFO304<sub>x</sub>（ $x=1, 2, \dots, 18$ ）は、構成行列の行数および列数である5つの枝に対応するメッセージを同時に読み出しもしくは書き込むことができるようになっている。

#### 【0126】

FIFO304<sub>1</sub>には、図17の検査行列の第1列目から第5列目までの1の位置に対応するデータが、各列共に縦方向に詰めた形に（0を無視した形で）格納される。すなわち、FIFO304<sub>1</sub>の第1の要素（第1段）には、検査行列の(1,1)から(5,5)の5×5の単位行列の1の位置に対応するデータが格納される。第2の要素には、検査行列の(6,1)から(10,5)の和行列（5×5の単位行列を右に1つだけサイクリックシフトした第1のシフト行列と、右に2つだけサイクリックシフトした第2のシフト行列との和である和行列）を構成する第1のシフト行列の1の位置に対応するデータが格納される。また、第3の要素には、検査行列の(6,1)から(10,5)の和行列を構成する第2のシフト行列の1の位置に対応するデータが格納される。

#### 【0127】

即ち、重みが2以上の構成行列については、その構成行列を、重みが1である

P×Pの単位行列、そのコンポーネントである1のうち1個以上が0になった準単位行列、または単位行列もしくは準単位行列をサイクリックシフトしたシフト行列のうちの複数の和の形で表現したときの、その重みが1の単位行列、準単位行列、またはシフト行列の1の位置に対応するデータ（単位行列、準単位行列、またはシフト行列に属する枝に対応するメッセージ）は、同一アドレス（FIFO 3 0 4<sub>1</sub>乃至3 0 4<sub>18</sub>のうちの同一のFIFO）に格納される。

#### 【0 1 2 8】

以下、第4および第5の要素についても、検査行列に対応づけて、データが格納される。このFIFO 3 0 4<sub>1</sub>の要素数（段数）は、検査行列の第1列から第5列における行方向の1の数（ハミング重み）の最大数である5になっている。

#### 【0 1 2 9】

FIFO 3 0 4<sub>2</sub>と3 0 4<sub>3</sub>も同様に検査行列に対応づけてデータを格納し、それぞれの長さ（段数）は、5である。FIFO 3 0 4<sub>4</sub>乃至3 0 4<sub>12</sub>も同様に検査行列に対応づけてデータを格納し、それぞれの長さは3である。FIFO 3 0 4<sub>13</sub>乃至3 0 4<sub>18</sub>も同様に検査行列に対応づけてデータを格納し、それぞれの長さは2である。但し、FIFO 3 0 4<sub>18</sub>の第1の要素は、検査行列の(1, 86)から(5, 90)に相当し、第5列目（検査行列の(1, 90)から(5, 90)）に1がないため、データは格納されない。

#### 【0 1 3 0】

以下、図18の復号装置の各部の動作について詳細に説明する。

#### 【0 1 3 1】

枝データ格納用メモリ3 0 0は、6つのFIFO 3 0 0<sub>1</sub>乃至3 0 0<sub>6</sub>からなり、前段のサイクリックシフト回路3 0 8から供給される5つのメッセージデータD311が、検査行列どの行に属するかの情報（Matrixデータ）D312に従って、データを格納するFIFOを、FIFO 3 0 0<sub>1</sub>乃至3 0 0<sub>6</sub>の中から選び、選んだFIFOに5つのメッセージデータD311をまとめて順番に格納していく。また、枝データ格納用メモリ3 0 0は、データを読み出す際には、FIFO 3 0 0<sub>1</sub>から5つのメッセージデータD300<sub>1</sub>を順番に読み出し、次段のセレクタ3 0 1に供給する。枝データ格納用メモリ3 0 0は、FIFO 3 0 0<sub>1</sub>からのメッセージデータの読み出しの終了後、FIF

03002乃至3006からも、順番に、メッセージデータを読み出し、セクタ301に供給する。

#### 【0132】

セクタ301は、セレクト信号D301に従って、FIFO3001乃至3006のうちの、現在データが読み出されているFIFOからの5つのメッセージデータを選択し、メッセージデータD302として、チェックノード計算部302に供給する。

#### 【0133】

チェックノード計算部302は、5つのチェックノード計算器3021乃至3025からなり、セクタ301を通して供給されるメッセージD302(D3021乃至D3025)を用いて、式(7)に従って演算を行い、その演算の結果得られる5つのメッセージD303(D3031乃至D3035)をサイクリックシフト回路303に供給する。

#### 【0134】

ここで、チェックノード計算器3021乃至3025それぞれは、図10に示したチェックノード計算器101と同様に構成される。

#### 【0135】

サイクリックシフト回路303は、チェックノード計算部302で計算された5つのメッセージD3031乃至D3035を、対応する枝が検査行列において元となる単位行列を幾つサイクリックシフトしたものであるかの情報(Matrixデータ) D305を元にサイクリックシフトし、その結果をメッセージD304として、枝データ格納用メモリ304に供給する。

#### 【0136】

枝データ格納用メモリ304は、18個のFIFO3041乃至30418からなり、前段のサイクリックシフト回路303から供給される5つのメッセージデータD304が検査行列のどの行に属するかの情報D305に従って、データを格納するFIFOを、FIFO3041乃至30418の中から選び、選んだFIFOに5つのメッセージデータD304をまとめて順番に格納していく。また、枝データ格納用メモリ304は、データを読み出す際には、FIFO3041から5つのメッセージD3061を順番に読み出し、次段のセクタ305に供給する。枝データ格納用メモリ304は、FIFO3041からのデータの読み出しの終了後、FIFO3042乃至30418からも、

順番に、メッセージデータを読み出し、セクタ305に供給する。

#### 【0137】

セクタ305は、セレクト信号D307に従って、FIFO304<sub>1</sub>乃至304<sub>18</sub>のうちの、現在データが読み出されているFIFOからの5つのメッセージデータを選択し、メッセージデータD308として、バリアブルノード計算部307と復号語計算部309に供給する。

#### 【0138】

一方、受信データ並べ替え部310は、通信路を通して受信したLDPC符号の符号系列（受信データ）D313を、式（9）の列置換を行うことにより並べ替え、符号系列D314として、受信用データメモリ306に供給する。受信データ用メモリ306は、受信データ並べ替え部310から供給される符号系列D314から、受信LLR（対数尤度比）を計算しており、その計算した受信LLRを5つまとめてデータD309として、バリアブルノード計算部307と復号語計算部309に供給する。

#### 【0139】

バリアブルノード計算部307は、5つのバリアブルノード計算器307<sub>1</sub>乃至307<sub>5</sub>からなり、セクタ305を通して供給されるメッセージD308(D308<sub>1</sub>乃至D308<sub>5</sub>)と、受信データ用メモリ306から供給される5つの受信LLR D309を用いて、式（1）に従って演算を行い、その演算の結果得られるメッセージD310(D310<sub>1</sub>乃至D310<sub>5</sub>)を、サイクリックシフト回路308に供給する。

#### 【0140】

ここで、バリアブルノード計算器307<sub>1</sub>乃至307<sub>5</sub>それぞれは、図11のバリアブルノード計算器103と同様に構成される。

#### 【0141】

サイクリックシフト回路308は、バリアブルノード計算部307で計算されたメッセージD310<sub>1</sub>乃至D310<sub>5</sub>を、対応する枝が検査行列において元となる単位行列を幾つサイクリックシフトしたものであるかの情報を元にサイクリックシフトし、その結果をメッセージD311として、枝データ格納用メモリ300に供給する。

#### 【0142】

以上の動作を1巡することで、LDPC符号の1回の復号を行うことができる。図18の復号装置は、所定の回数だけLDPC符号を復号した後、復号語計算部309および復号データ並べ替え部311において、最終的な復号結果を求めて出力する。

#### 【0143】

即ち、復号語計算部309は、5つの復号語計算器309<sub>1</sub>乃至309<sub>5</sub>からなり、セクタ305が出力する5つのメッセージD308(D308<sub>1</sub>乃至D308<sub>5</sub>)と、受信データ用メモリ306から供給される5つの受信LLR D309を用い、複数回の復号の最終段において、式(5)に基づいて、復号結果(復号語)を計算して、その結果得られる復号データD315を、復号データ並べ替え部311に供給する。

#### 【0144】

復号データ並べ替え部311は、復号語計算部309から供給される復号データD315を対象に、式(9)の列置換の逆置換を行うことにより、その順序を並べ替え、最終的な復号結果D316として出力する。

#### 【0145】

なお、枝データ(枝に対応するメッセージ)が欠けている箇所に関しては、メモリ格納時(枝データ格納用メモリ300と304へのデータ格納時)には、何のメッセージも格納せず、また、ノード演算時(チェックノード計算部302でのチェックノード演算時とバリアブルノード計算部307でのバリアブルノード演算時)にも何の演算も行わない。

#### 【0146】

また、サイクリックシフト回路303および308には、バレルシフトを用いると回路規模を小さくしながら所望の操作を実現できる。

#### 【0147】

上記説明には、枝データ格納にFIFOを用いたが(枝データ格納メモリ300と304をFIFOで構成するようにしたが)、FIFOの代わりにRAMを用いても構わない。その場合、RAMには、P個の枝情報(枝に対応するメッセージ)を同時に読み出すことの出来るビット幅と、枝総数/Pのワード(word)数が必要となる。さらに、RAMへの書き込みは、検査行列の情報から、書き込もうとしているデータが次

に読み出される際に何番目に読み出されるかを求め、その位置に書き込む。また、RAMからの読み出しの際には、アドレスの先頭から順次データを読み出す。FIFOの代わりにRAMを用いると、セクタ301および305は不要になる。

#### 【0148】

なお、FIFOやRAMの物理的なビット幅が足りない場合には、複数のRAMを用いて同じ制御信号を与えることで、論理的に1つのRAMとみなすことができる。

#### 【0149】

また、上述の場合には、説明を簡単にするために、Pが5の場合、即ち、検査行列を構成する構成行列の行数および列数が5の場合を例に挙げたが、構成行列の行数および列数Pは必ずしも5である必要はなく、検査行列によって異なる値を取ることもあり得る。例えば、Pは360や392であつてもよい。

#### 【0150】

また、本実施の形態では、符号長90、符号化率2/3のLDPC符号を用いたが、LDPC符号の符号長や符号化率は、幾つであつても構わない。例えば、構成行列の行数および列数Pが5の場合、枝総数が5以下であれば、どんな符号長、符号化率のLDPC符号でも、制御信号を代えるだけで、図18の復号装置を用いて復号可能である。

#### 【0151】

さらに、構成行列の行数および列数Pが所定の値で、枝の総数がある値以下、という条件を満たすあるLDPC符号の復号装置は、その条件を満たす、任意の符号長で、任意の符号化率のLDPC符号を復号することができる。

#### 【0152】

以上のように、検査行列（元の検査行列）に対して、行置換と列置換うちの一方または両方を施し、 $P \times P$ の単位行列、そのコンポーネントの1のうち1個以上が0になった準単位行列、単位行列もしくは準単位行列をサイクリックシフトしたシフト行列、単位行列、準単位行列、もしくはシフト行列の複数の和である和行列、 $P \times P$ の0行列の組合せ、つまり、構成行列の組み合わせで表わすことができる検査行列（変換検査行列）に変換することで、LDPC符号の復号を、チェックノードとバリアブルノードの演算をP個同時に行うアーキテクチャ(architecture

)を採用することが可能となり、これにより、ノード演算を、P個同時に行うことで動作周波数を実現可能な範囲に抑えることができ、多数の繰り返し復号を行うことを可能にしつつ、メモリ（FIFOやRAM）への書き込みと読み出し時に、異なるアドレスへの同時アクセスが起きることを防止することができる。

#### 【0153】

即ち、上述した検査行列の行置換と列置換の双方もしくは片方を行うことで、チェックノードとバリエブルノードの演算をP個同時に行うことが可能となり、さらに、このようにノード演算をP個同時に行うことで、動作周波数を実現可能な範囲に抑えることができ、多数の繰り返し復号を行うことを可能にしつつ、メモリ（FIFOやRAM）への書き込みと読み出し時に、異なるアドレスへの同時アクセスが起きることを防止することができる。

#### 【0154】

さらに、図17の検査行列（変換検査行列）で表わされるLDPC符号を復号する場合には、269個の枝をチェックノード、バリエブルノード毎に5個ずつ演算することが可能であることから、1回の復号に、 $269/5 \times 2 \div 108$ クロック動作すればよいことになる。50回の復号には、90個の符号情報を受信する間に、 $108 \times 50 = 5400$ クロック動作すればよいことになり、受信周波数の約60倍の動作周波数でよいことになる。従って、図18の復号装置によれば、各ノード演算を一つずつ行う図9の復号装置に比べて、1/5の動作周波数で済むことになる。また、回路規模の面から見ても、メモリの大きさは同じであるため、論理回路が多少大きくなっても全体への影響は小さいと言える。

#### 【0155】

一般的に、LDPC符号は符号長が数千から数万と大きいため、Pの値も数百の大きさを持つものが使われる。その場合には、更に本発明に係る復号装置を用いる効果は大きくなる。

#### 【0156】

以上のように、ロジック、メモリ共に回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑えることができ、メモリアクセスの制御も容易に行うことができる。



## 【0157】

また、LDPC符号の性質から、検査行列に行置換や列置換を施しても符号の性能は変わらない。従って、行置換または列置換によって、構成行列の組み合わせで表すことができる変換検査行列を得られる検査行列に対応するLDPC符号については、どのような符号長、符号化率のLDPC符号であっても、性能の劣化を招くことなく、実装が容易で、かつ効率の良い復号を行うことができる。

## 【0158】

さらに、本発明に係る復号装置は、サムプロダクトアルゴリズムを忠実に実装するものであるため、メッセージの量子化以外の復号損失が起きることはない。

## 【0159】

以上の観点から、本発明に係る復号装置を用いることで、高性能な復号が可能になる。

## 【0160】

なお、検査行列が、構成行列の行数および列数Pの倍数でない場合は、検査行列の端数の外側にすべて0 (all 0) の成分を付けてPの倍数とみなして適用できることがある。

## 【0161】

次に、上述した一連の処理は、ハードウェアにより行うこともできるし、ソフトウェアにより行うこともできる。一連の処理をソフトウェアによって行う場合には、そのソフトウェアを構成するプログラムが、汎用のコンピュータ等にインストールされる。

## 【0162】

そこで、図19は、上述した一連の処理を実行するプログラムがインストールされるコンピュータの一実施の形態の構成例を示している。

## 【0163】

プログラムは、コンピュータに内蔵されている記録媒体としてのハードディスク405やROM403に予め記録しておくことができる。

## 【0164】

あるいはまた、プログラムは、フレキシブルディスク、CD-ROM(Compact Disc

Read Only Memory), MO(Magneto Optical)ディスク, DVD(Digital Versatile Disc)、磁気ディスク、半導体メモリなどのリムーバブル記録媒体 411 に、一時的あるいは永続的に格納(記録)しておくことができる。このようなリムーバブル記録媒体 411 は、いわゆるパッケージソフトウェアとして提供することができる。

#### 【0165】

なお、プログラムは、上述したようなリムーバブル記録媒体 411 からコンピュータにインストールする他、ダウンロードサイトから、デジタル衛星放送用の人工衛星を介して、コンピュータに無線で転送したり、LAN(Local Area Network)、インターネットといったネットワークを介して、コンピュータに有線で転送し、コンピュータでは、そのようにして転送されてくるプログラムを、通信部 408 で受信し、内蔵するハードディスク 405 にインストールすることができる。

#### 【0166】

コンピュータは、CPU(Central Processing Unit) 402 を内蔵している。CPU 402 には、バス 401 を介して、入出力インタフェース 410 が接続されており、CPU 402 は、入出力インタフェース 410 を介して、ユーザによって、キーボードや、マウス、マイク等で構成される入力部 407 が操作等されることにより指令が入力されると、それにしたがって、ROM(Read Only Memory) 403 に格納されているプログラムを実行する。あるいは、また、CPU 402 は、ハードディスク 405 に格納されているプログラム、衛星若しくはネットワークから転送され、通信部 408 で受信されてハードディスク 405 にインストールされたプログラム、またはドライブ 409 に装着されたリムーバブル記録媒体 411 から読み出されてハードディスク 405 にインストールされたプログラムを、RAM(Random Access Memory) 404 にロードして実行する。これにより、CPU 402 は、上述したフローチャートにしたがった処理、あるいは上述したブロック図の構成により行われる処理を行う。そして、CPU 402 は、その処理結果を、必要に応じて、例えば、入出力インタフェース 410 を介して、LCD(Liquid Crystal Display)やスピーカ等で構成される出力部 406 から出力、あるいは、通信部 4

0 8 から送信、さらには、ハードディスク 4 0 5 に記録等させる。

【0 1 6 7】

ここで、本明細書において、コンピュータに各種の処理を行わせるためのプログラムを記述する処理ステップは、必ずしもフローチャートとして記載された順序に沿って時系列に処理する必要はなく、並列的あるいは個別に実行される処理（例えば、並列処理あるいはオブジェクトによる処理）も含むものである。

【0 1 6 8】

また、プログラムは、1 のコンピュータにより処理されるものであっても良いし、複数のコンピュータによって分散処理されるものであっても良い。さらに、プログラムは、遠方のコンピュータに転送されて実行されるものであっても良い。

【0 1 6 9】

【発明の効果】

以上の如く、本発明によれば、回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行うことが可能となる。

【図面の簡単な説明】

【図 1】

LDPC符号の検査行列Hを説明する図である。

【図 2】

LDPC符号の復号手順を説明するフローチャートである。

【図 3】

メッセージの流れを説明する図である。

【図 4】

LDPC符号の検査行列の例を示す図である。

【図 5】

検査行列のタナグラフを示す図である。

【図 6】

バリエابلノードを示す図である。

【図 7】

チェックノードを示す図である。

【図 8】

LDPC符号の検査行列の例を示す図である。

【図 9】

ノード演算を一つずつ行うLDPC符号の復号装置の構成例を示すブロック図である。

【図 10】

メッセージを一つずつ計算するチェックノード計算器の構成例を示すブロック図である。

【図 11】

メッセージを一つずつ計算するバリアブルノード計算器の構成例を示すブロック図である。

【図 12】

ノード演算を全て同時に行うLDPC符号の復号装置の構成例を示すブロック図である。

【図 13】

メッセージを同時に計算するチェックノード計算器の構成例を示すブロック図である。

【図 14】

メッセージを同時に計算するバリアブルノード計算器の構成例を示すブロック図である。

【図 15】

LDPC符号の検査行列の例を示す図である。

【図 16】

検査行列に行置換と列置換を施した行列を示す図である。

【図 17】

5×5単位に分割した検査行列を示す図である。

【図 18】

本発明を適用した復号装置の一実施の形態の構成例を示すブロック図である。

## 【図19】

本発明を適用したコンピュータの一実施の形態の構成例を示すブロック図である。

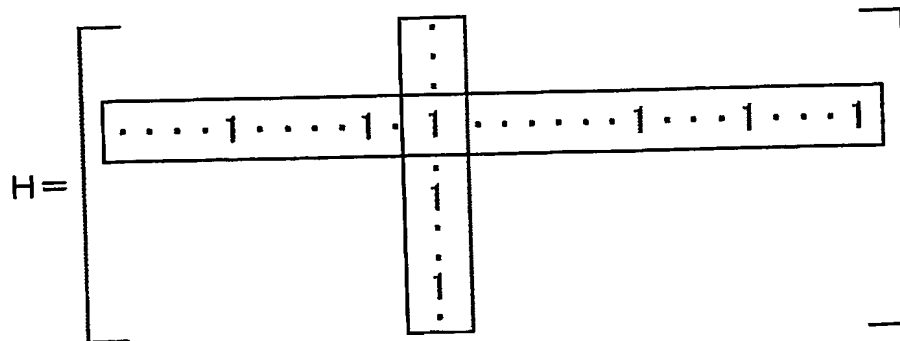
## 【符号の説明】

300 枝データ格納用メモリ, 301 セレクタ, 302 チェックノード計算部, 303 サイクリックシフト回路, 304 枝データ格納用メモリ, 305 セレクタ, 306 受信データ用メモリ, 307 バリアブルノード計算部, 308 サイクリックシフト回路, 309 復号語計算部, 310 受信データ並べ替え部, 311 復号データ並べ替え部, 401 バス, 402 CPU, 403 ROM, 404 RAM, 405 ハードディスク, 406 出力部, 407 入力部, 408 通信部, 409 ドライブ, 410 入出力インタフェース, 411 リムーバブル記録媒体

【書類名】 図面

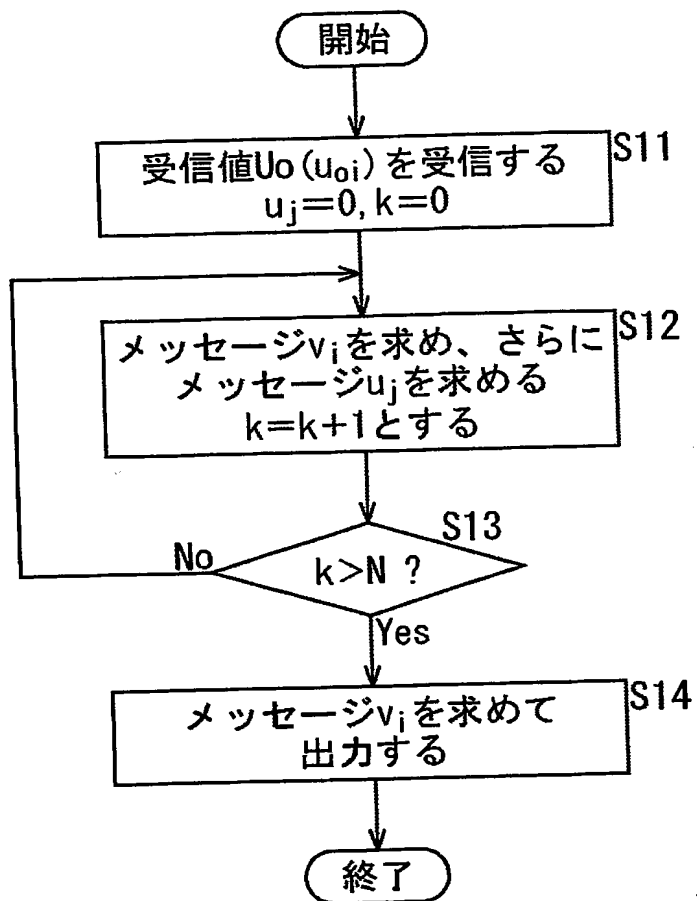
【図 1】

図1



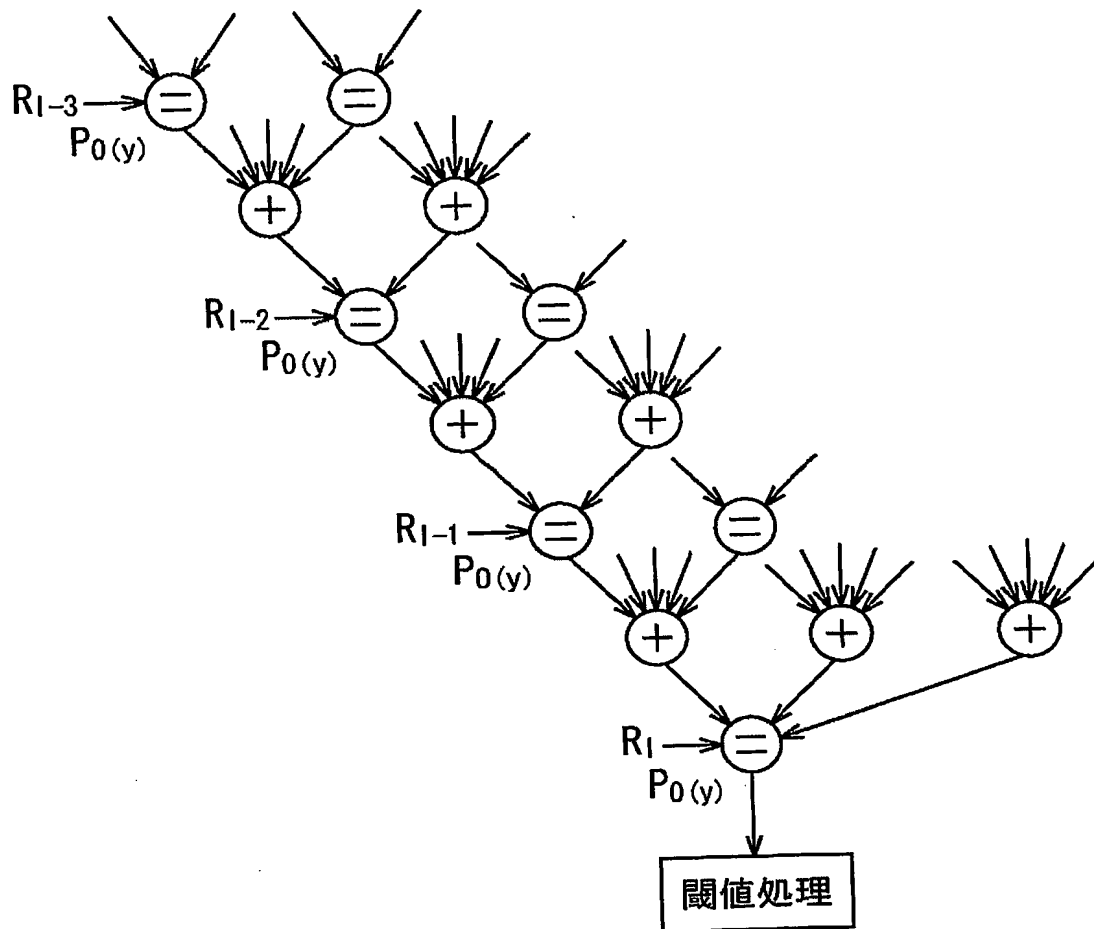
【図2】

図2



【図 3】

図3





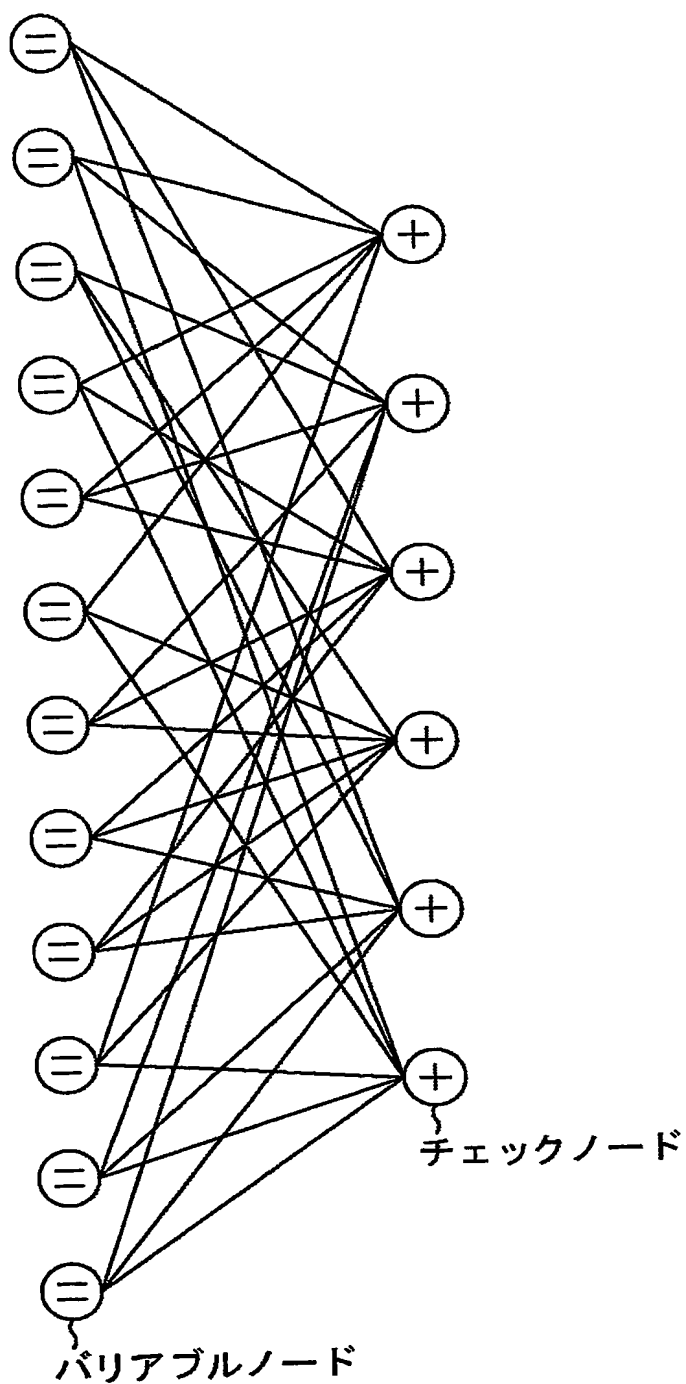
【図4】

図4

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

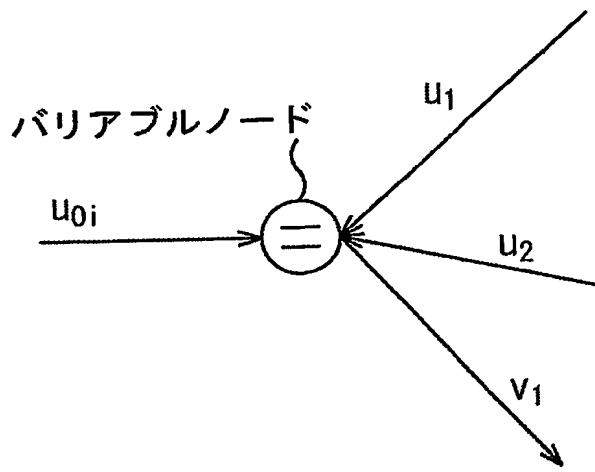
【図 5】

図5



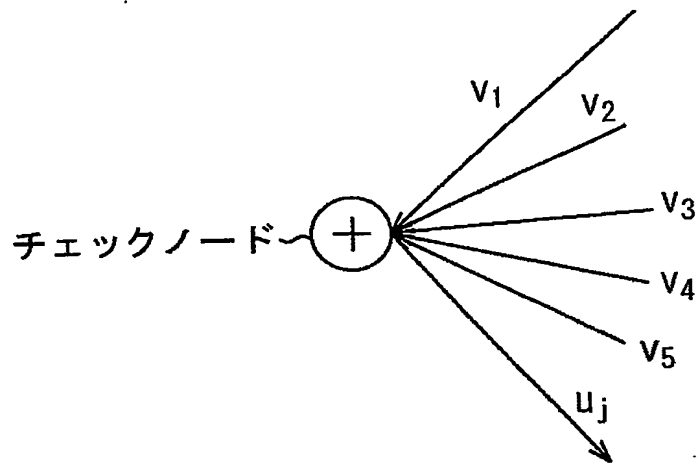
【図 6】

図6



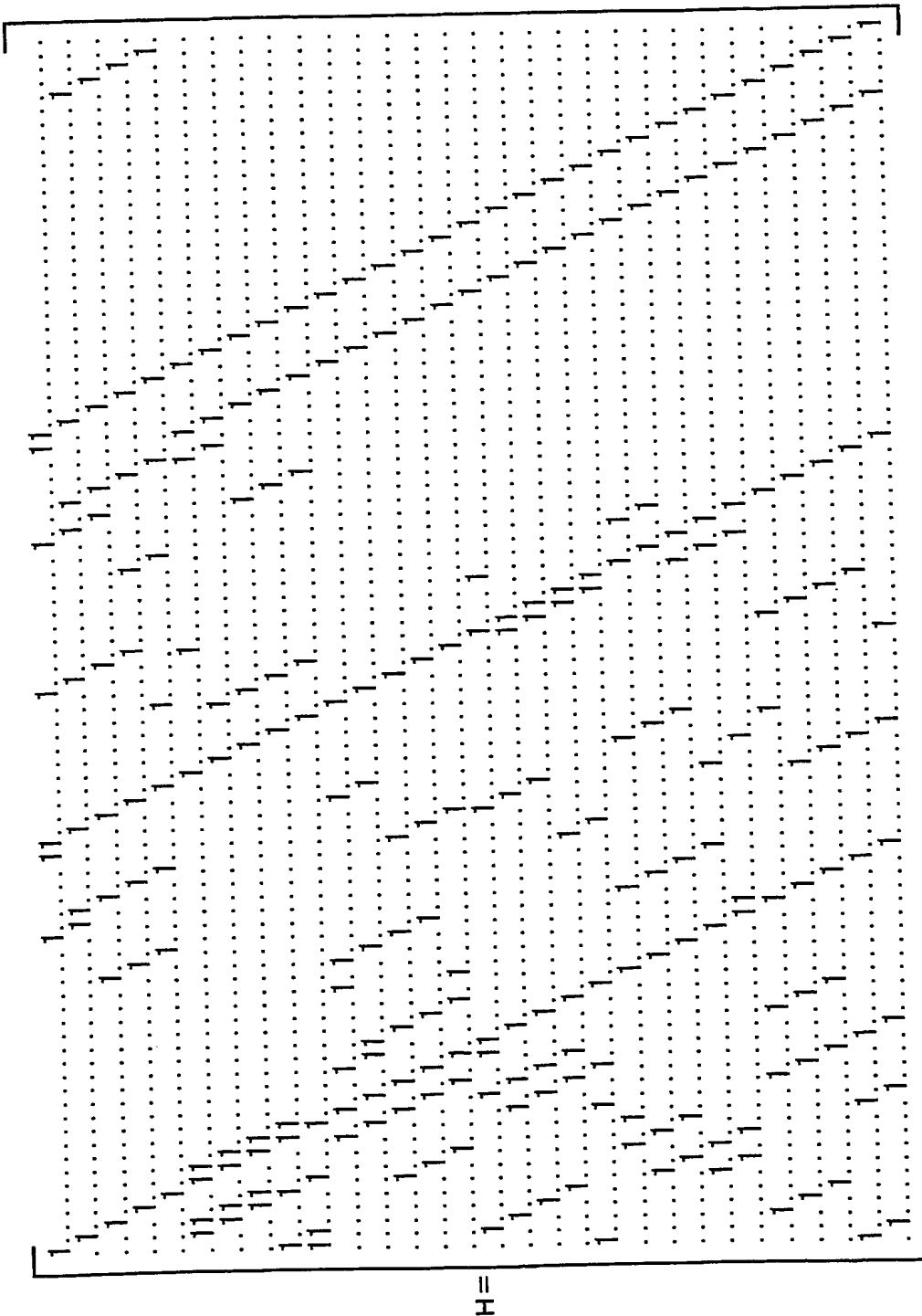
【図 7】

図 7



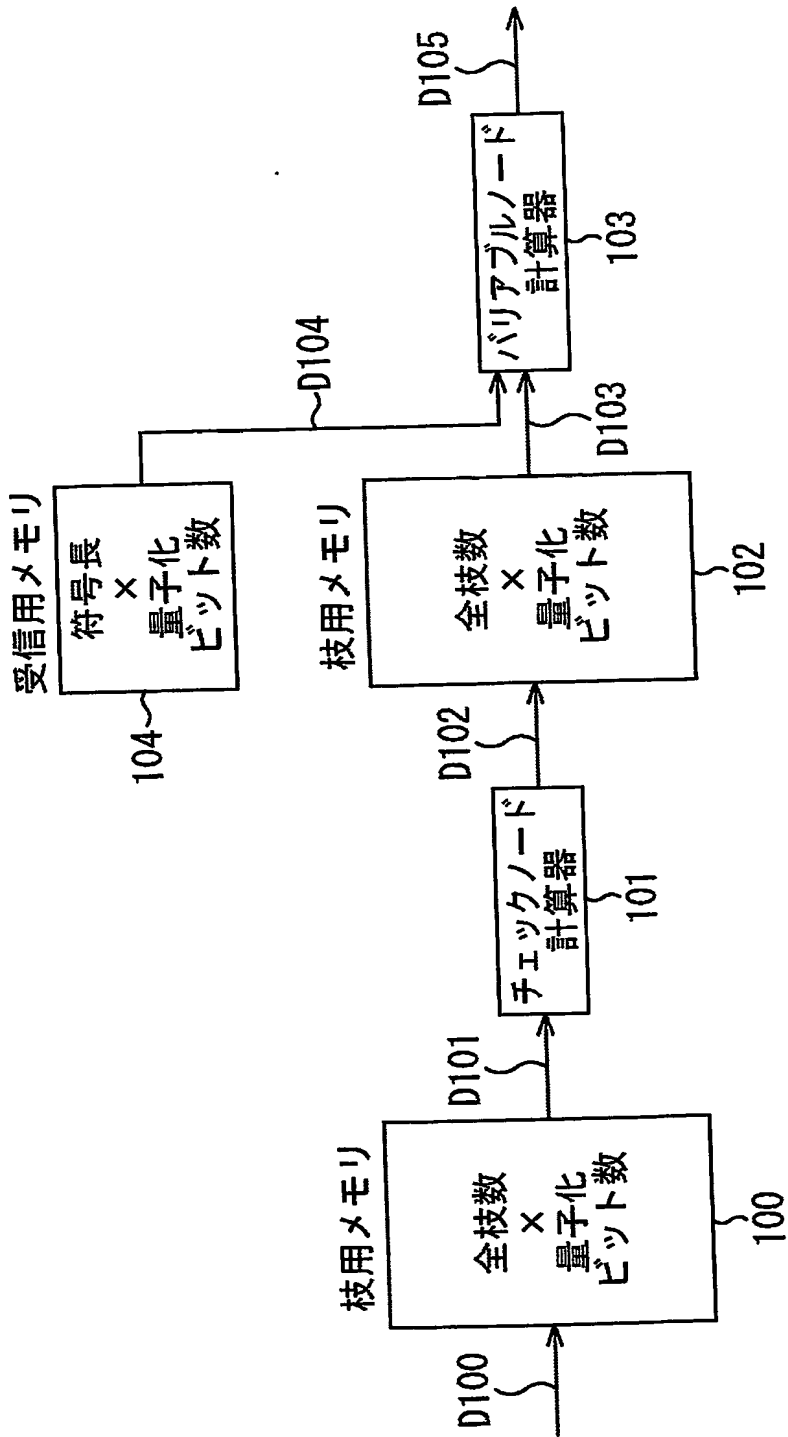
【図8】

図8



【図 9】

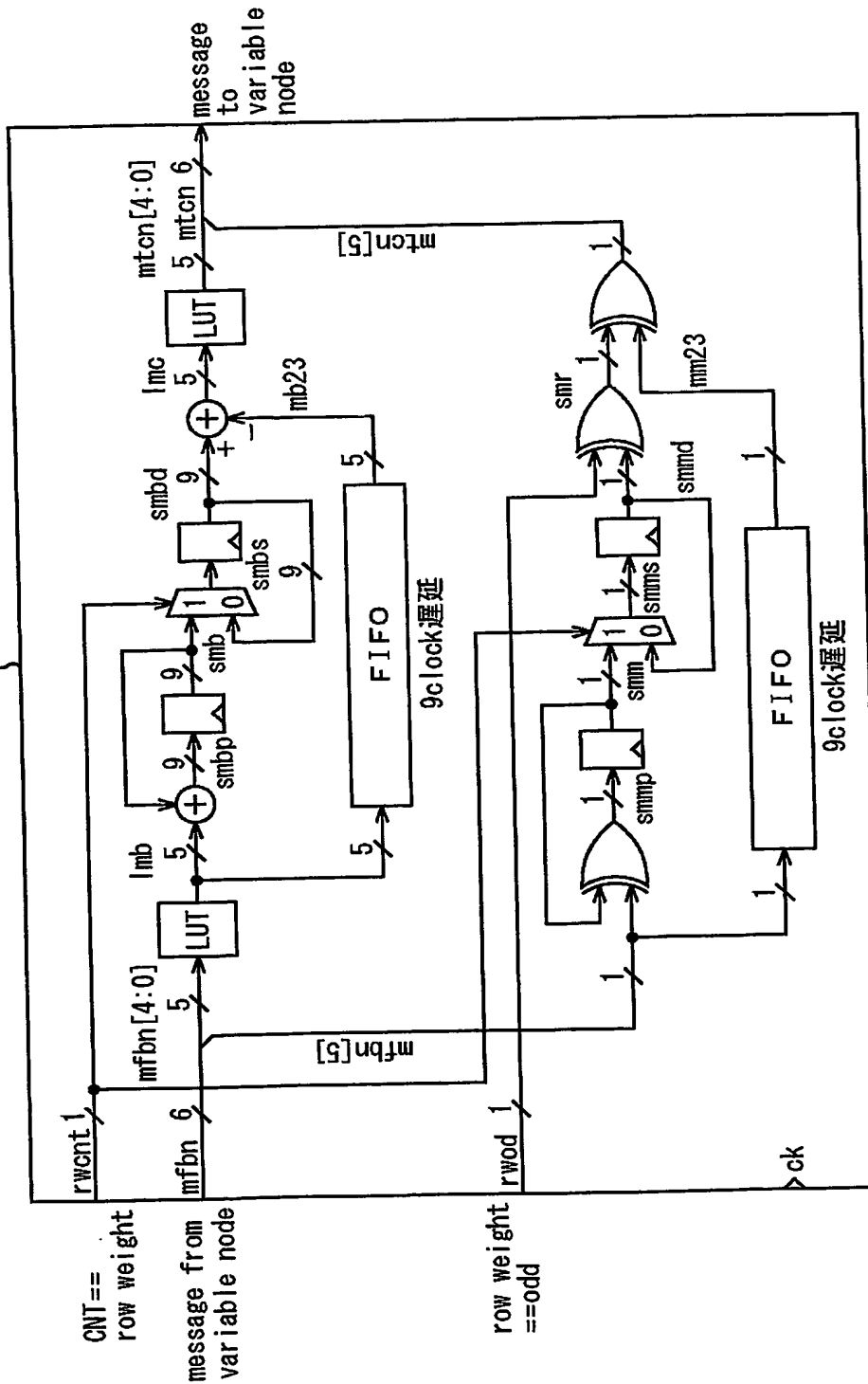
図9



【図 10】

図10

チェックノード計算器 101



【図 11】

图11

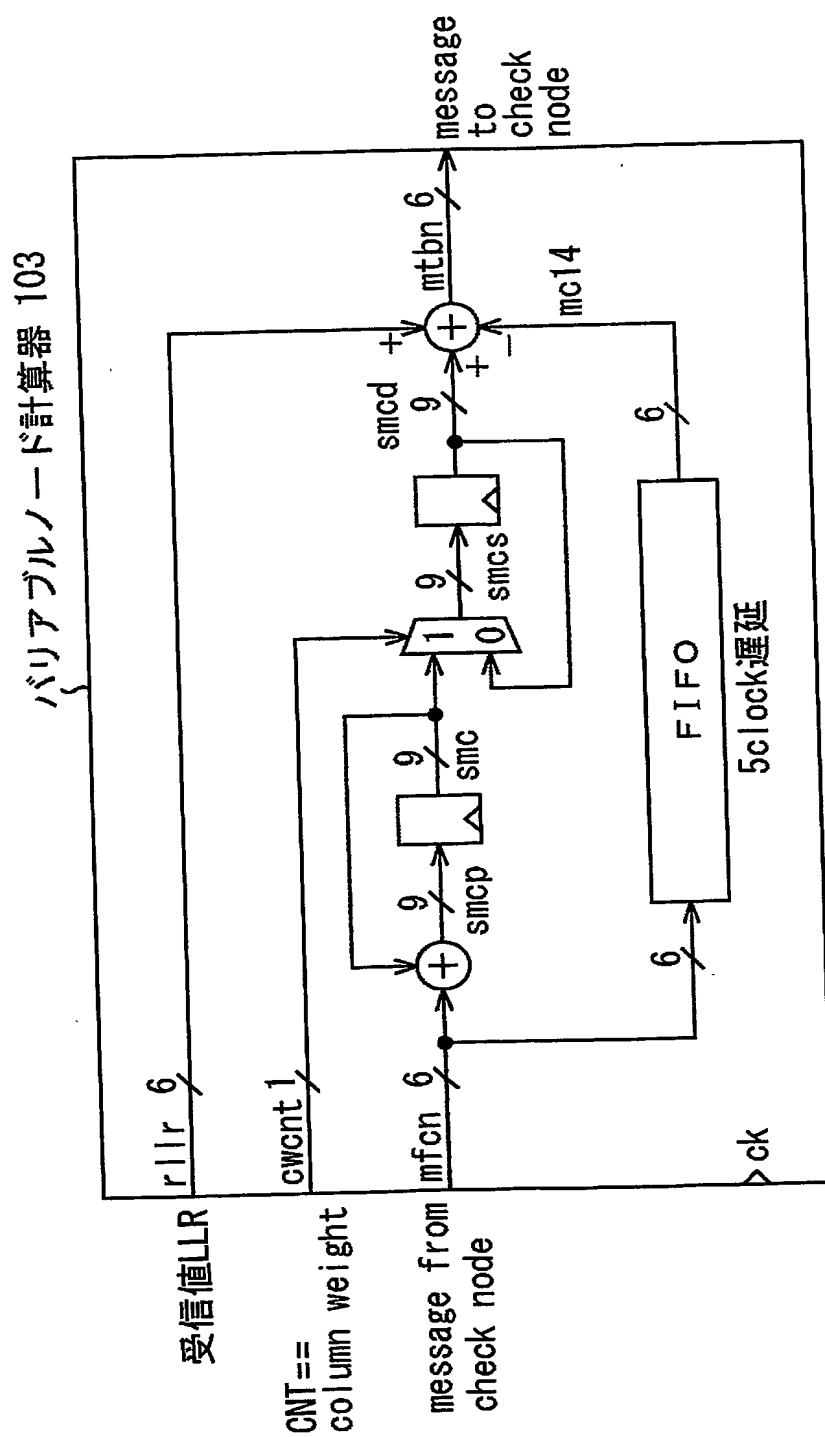
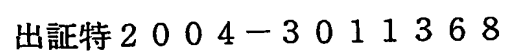


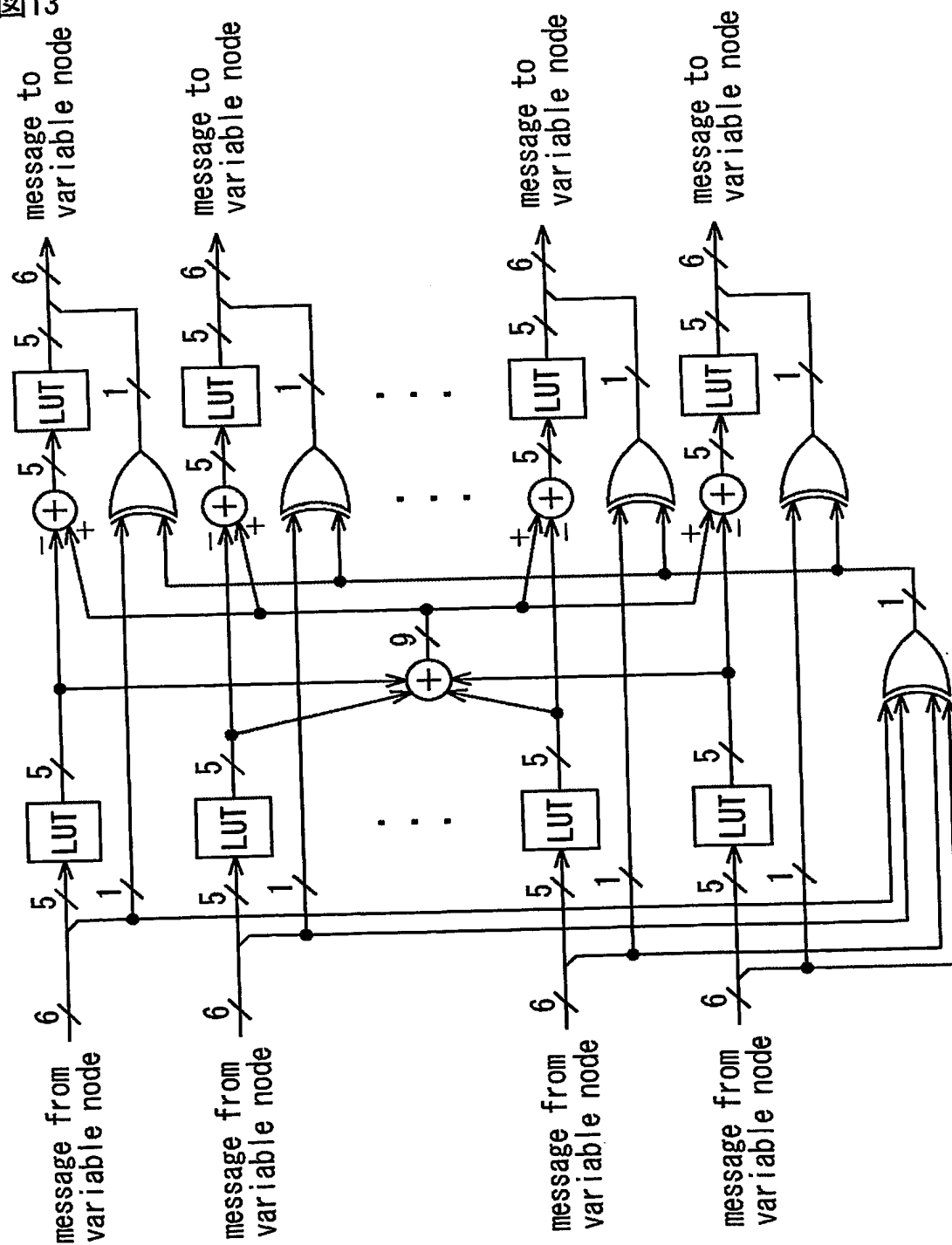


图12



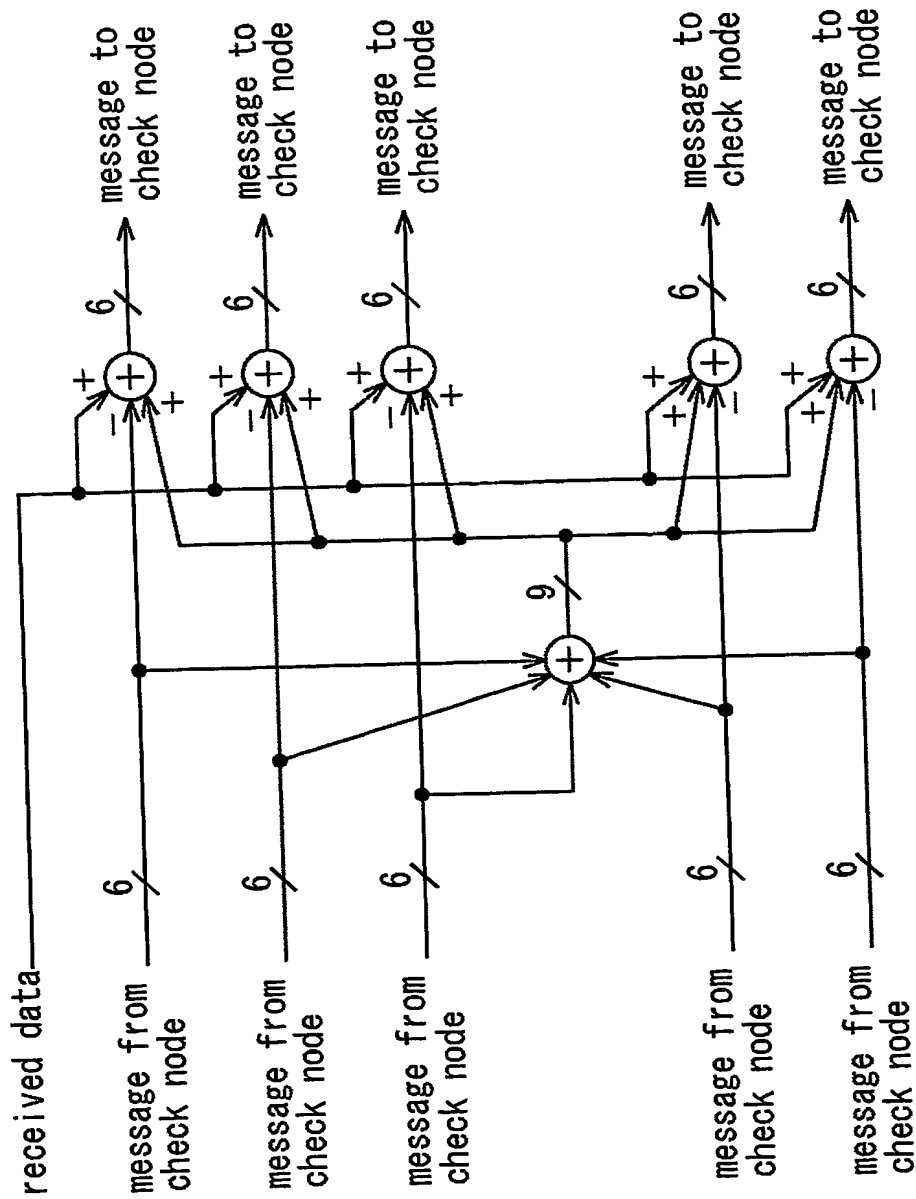
【図 13】

图13



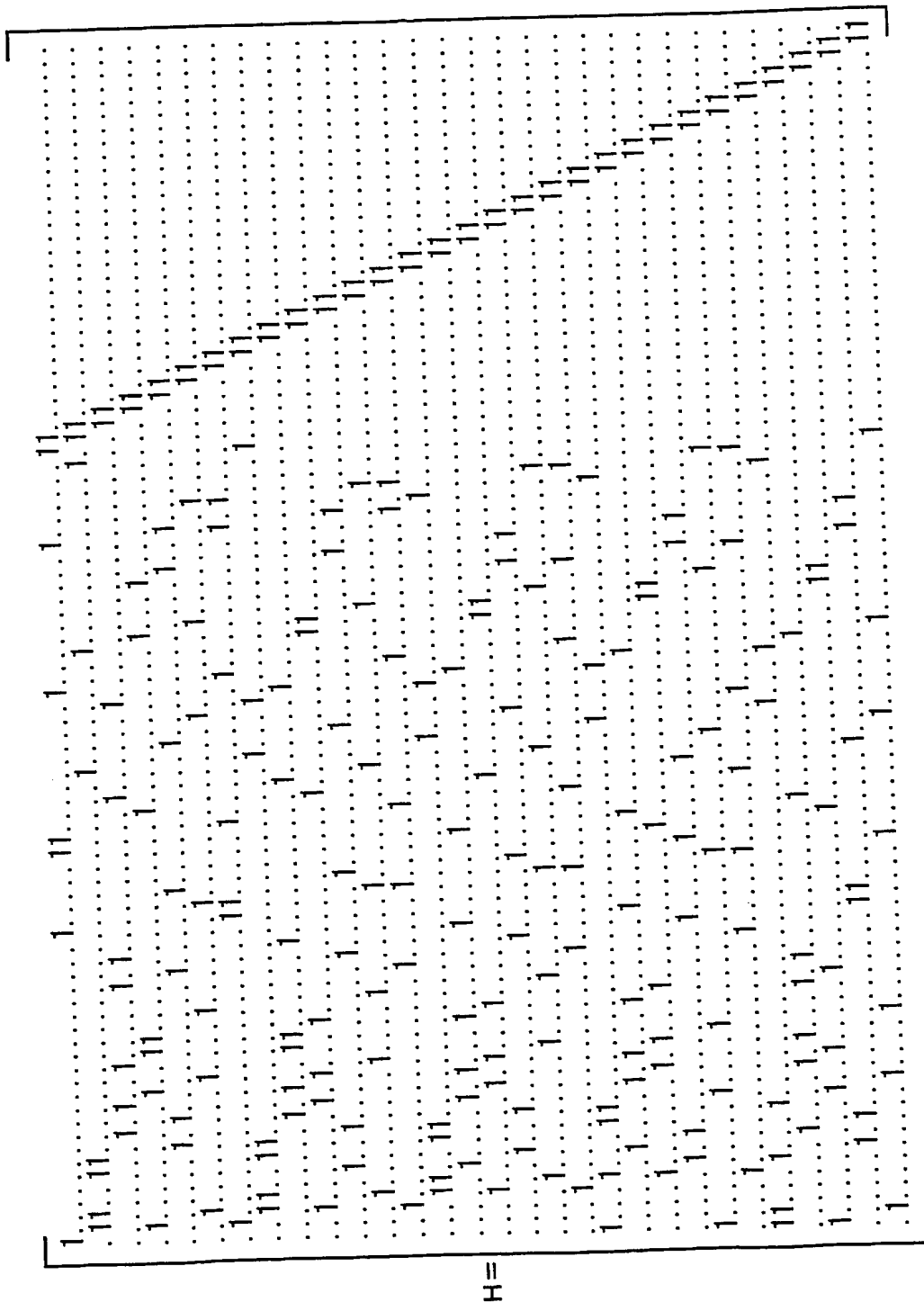
【図 14】

図14



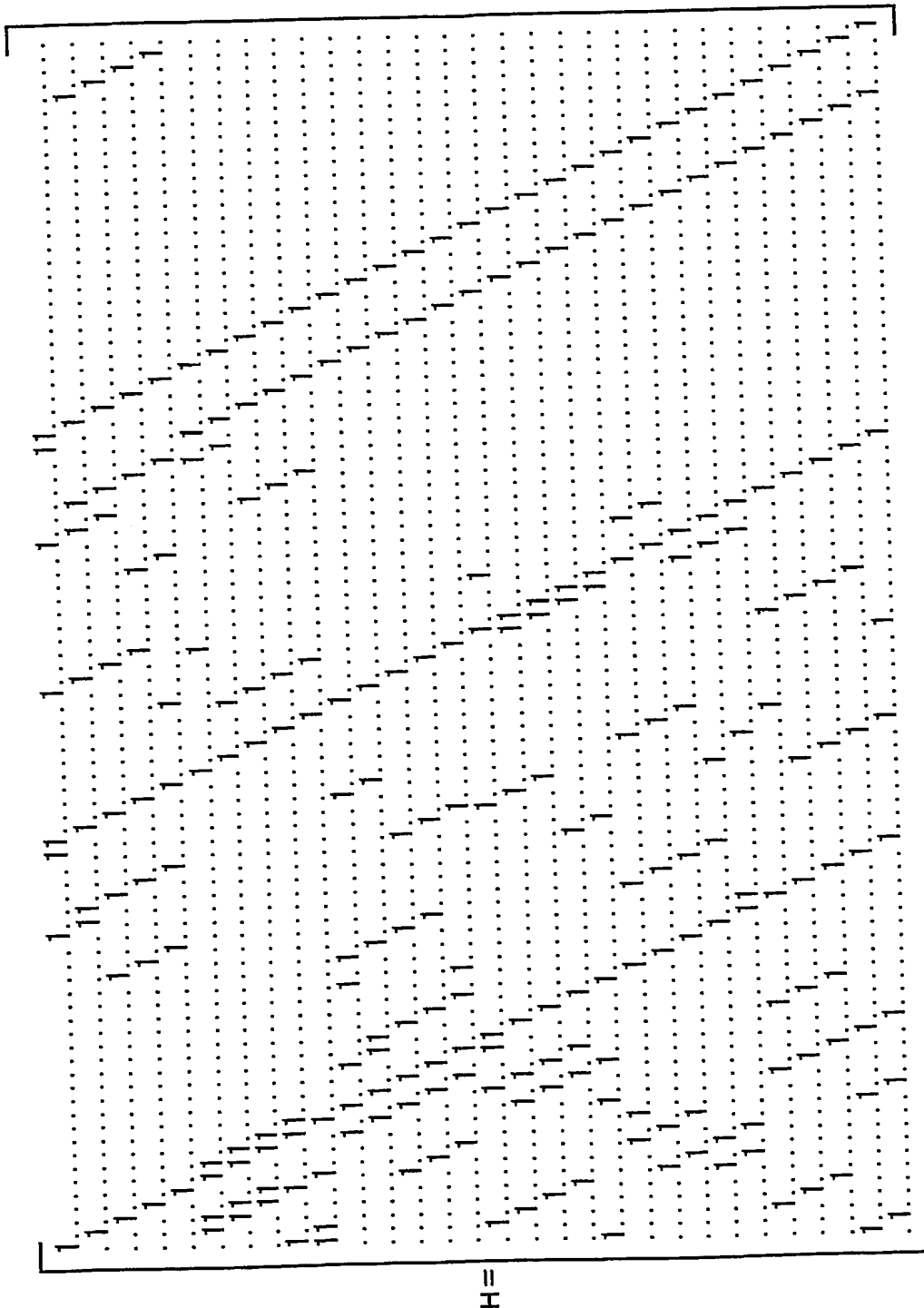
【図15】

図15



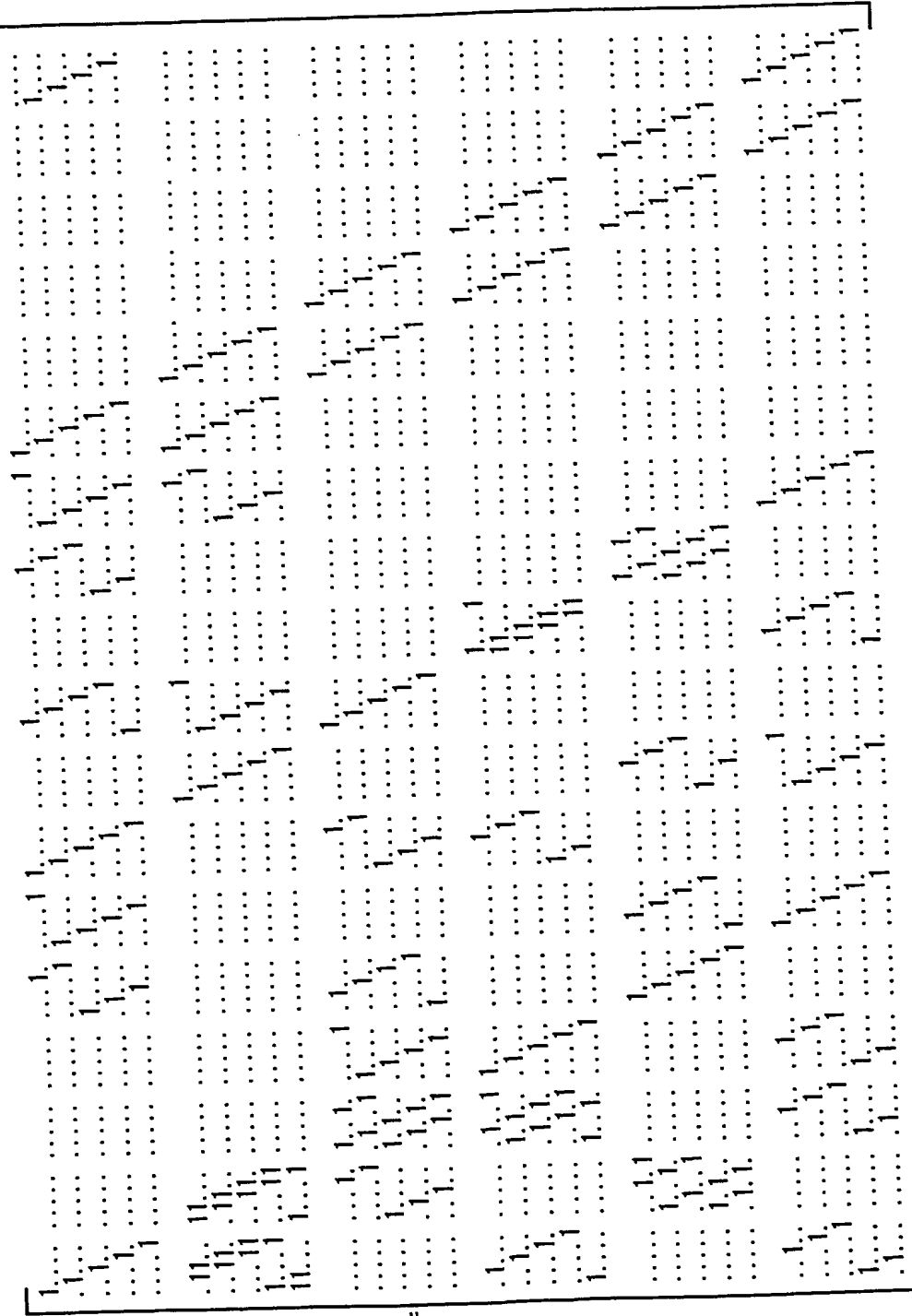
【図 1 6】

図16



【図17】

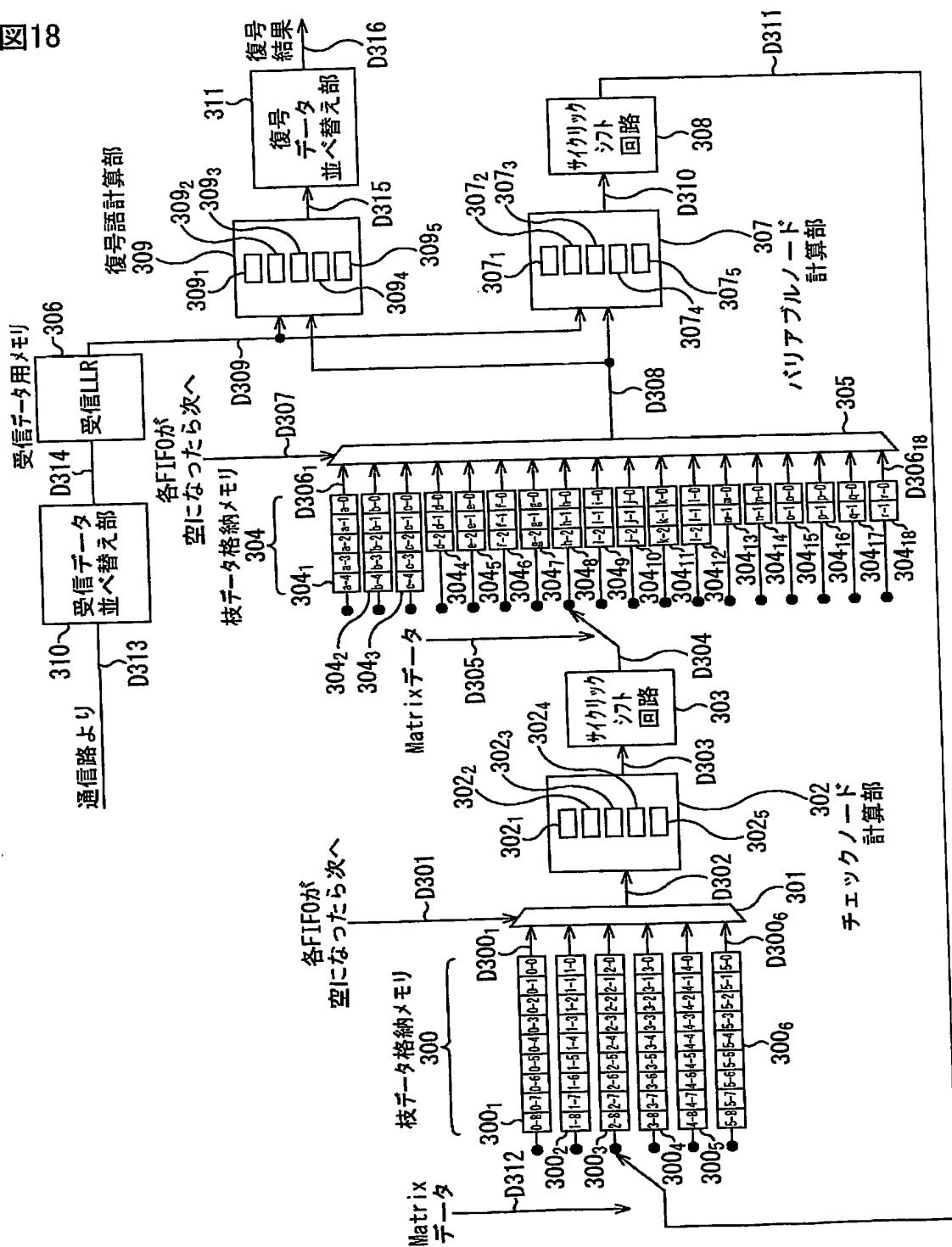
図17



||  
H

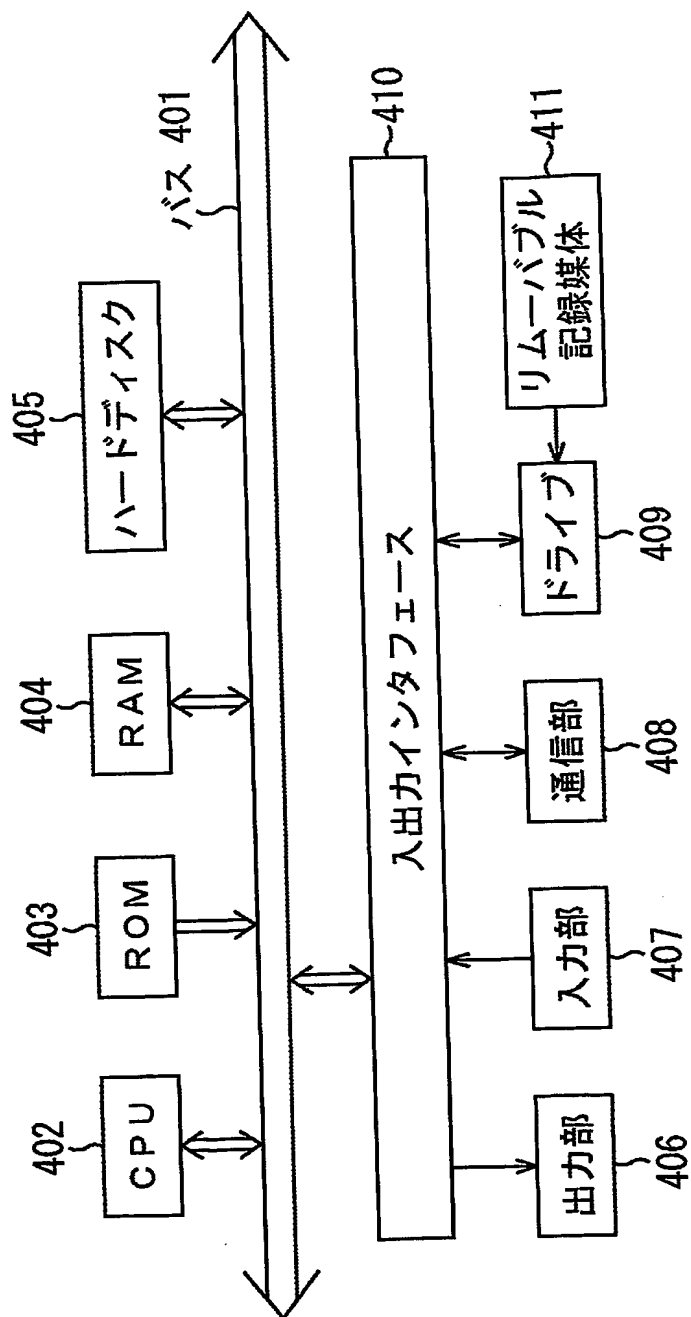
【図 18】

図 18



【図 19】

図19



コンピュータ



【書類名】 要約書

【要約】

【課題】 回路規模を抑制しつつ、動作周波数も十分実現可能な範囲に抑え、メモリアクセスの制御も容易に行う。

【解決手段】 LDPC (Low Density Parity Check) 符号の元の検査行列に対して、行置換と列置換のうちの一方または両方を行って得られる変換検査行列を用いて、LDPC符号が復号される。この場合に、 $P \times P$ の単位行列、その単位行列のコンポーネントである 1 のうちの 1 個以上が 0 になった行列である準単位行列、単位行列もしくは準単位行列をサイクリックシフトした行列であるシフト行列、単位行列、準単位行列、もしくはシフト行列のうちの複数の和である和行列、または  $P \times P$ の 0 行列を構成行列として、変換検査行列は、複数の構成行列の組合せで表される。チェックノード計算部 302 は、チェックノードの演算を、 $P$ 個同時に行い、バリアブルノード計算部 304 は、バリアブルノードの演算を、 $P$ 個同時に行う。

【選択図】 図 18

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**